Studies of Singular Value Decomposition in Terms of Integrable Systems(Dissertation_全文)

AUTHOR(S):

Iwasaki, Masashi

Studies of Singular Value Decomposition in Terms of Integrable Systems

Masashi Iwasaki

Studies of Singular Value Decomposition in Terms of Integrable Systems

Masashi Iwasaki

Division of Applied Mathematical Analysis
Department of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto University
Sakyo-ku Kyoto 606-8501 Japan

2004

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In this thesis, we study integrable systems and their applications to numerical algorithms for computing *singular value decomposition (SVD)*. We first prove that singular values are computable by using certain integrable systems. We design a numerical algorithm, named the *discrete Lotka-Volterra (dLV) algorithm*, for computing singular values. Next we explain several features of the dLV algorithm and propose a method for computing the corresponding singular vectors. Finally by introducing a shift of origin to dLV algorithm to accelerate the convergence we design a new efficient SVD algorithm with respect to both convergence speed and numerical accuracy.

1. Numerical algorithms for singular value decomposition in LAPACK

One of the most important decompositions in matrix computation is the SVD. For any rectangular matrix $A \in \boldsymbol{R}^{\ell \times m}$, there are orthogonal matrices $U \in \boldsymbol{R}^{\ell \times \ell}$ and $V \in \boldsymbol{R}^{m \times m}$ such that $U^\top AV$ holds $U^\top AV = (\Sigma\ O)^\top$ or $(\Sigma\ O)$, where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_p)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, $p = \min\{\ell, m\}$ and O is the zero matrix. Here σ_k, $k = 1, 2, \ldots, p$ are singular values of A. Here $A = U(\Sigma\ O)^\top V^\top$ or $U(\Sigma\ O)V^\top$ is just the SVD of A.

The SVD is a powerful technique dealing with certain equations or matrices that are either singular or numerically very close to singular. It allows us to comprehend problems related to a given rectangular matrix and provides numerical answer as well. Many times we encounter the SVD approch in the linear *Least Squares (LS) problem* to find a real m-vector x_0 minizing the euclidean length of $Ax - b$, where b is a given real ℓ-vector and the rank of A is less than $\min\{\ell, m\}$ [**20**]. The SVD is related to the LS problem and is particularly useful in analyzing the effect of data errors as they influence solutions to the LS problem. The LS problem is known by different names by those in scientific disciplines. Namely the SVD has wider application in many fields, for example, mathematics, numerical analysis, statistics, engineering and so on.

The practical SVD is indirectly utilized in a wide variety of domains, for example, the multivariate data analysis and the 3 dimensional (3D) reconstruction in the computer vision field. Now there are several demands for such computation of the SVD as fast computation, applicability to large-size problem, and high relative accuracy. In the multivariate data analysis, the SVD is used for the latent semantic indexing (LSI) [**3**]. The LSI takes a large matrix of term-document association data and constructs a semantic space. The SVD arranges the space to

reflect the major associative patterns in the data, and ignore the smaller, less important influences. The amount of the data on the WEB is increasing day by day. It requires fast computation of the large-size SVD problem. In the computer vision field, the 3D reconstruction technique recovers the 3D geometry from an 2D image sequence which is given by a matrix [**43**]. The SVD enables us to compute both shape and motion at the same instant by factoring the matrix into two matrices which represent objective shape and camera rotation, respectively. The 3D reconstruction has been used in robot vision and autonomous vehicles. The 3D geometry data deeply affects the stability of robots and vehicles with respect to their visual perceptions. It is essential for the SVD to compute stably with high relative accuracy in these cases.

Rutishauser improved his *quotient difference (qd) algorithm* to compute, for example, poles of a class of meromorphic functions. The original qd algorithm [**35**] takes a simple form and is free from any square root computation, however, it is not always stable. Therefore a progressive form of the *qd (pqd) algorithm* and its variant called the *differential qd (dqd) algorithm* were presented by himself. See the book [**38**] by Rutishauser and a survey paper [**32**] by Parlett for those improved qd algorithms.

It can be applied to a matrix eigenvalue problem for a tridiagonal matrix as the *LR* algorithm [**37**]. The pqd algorithm is backward stable when every qd variables are positive [**38**]. In this case the eigenvalues are all real, positive and simple (see [**38**], p.468). The dqd algorithm does not need any subtractions (see [**7**], p.198). Though the *QR* algorithm was found in 1961 as a stable variant of the qd algorithm [**33**], the qd algorithm itself has not occupied a major position in numerical linear algebra up to '90s.

In 1965, Golub and Kahan (see [**8**]) proposed an effective SVD algorithm consisting of two distinct processes. The first process is a transformation of any given rectangular matrix $A \in \boldsymbol{R}^{\ell \times m} (\ell \geq m)$ to an upper bidiagonal matrix $B \in \boldsymbol{R}^{m \times m}$ without changing singular values. The Householder transformation is adopted for this process. The second process of the Golub-Kahan SVD algorithm is performed by applying the *QR* algorithm to the symmetric positive tridiagonal matrix $B^{\top}B$. Then each singular value of A is given as the positive square root of each eigenvalue of $B^{\top}B$. An SVD of A in the case $\ell \leq m$ is also performed by the same process as the case $\ell \geq m$. Several results based on their idea have been found. Especially, the *QR* algorithm part is improved by Golub-Reinsch[**9**], Demmel-Kahan[**6**] and so on. Golub-Reinsch introduced a shift of origin into the *QR* algorithm. The Golub-Reinsch version computes an SVD of B much faster than the original *QR* algorithm. In 1990 Demmel-Kahan proposed a definitive version of the *QR* algorithm, and then were awarded the second SIAM prize in numerical linear algebra. The *Demmel-Kahan (DK) algorithm* requires many times of square root computation and computed singular value is in relative error less than $O(m^3\varepsilon)$, where m is the size of matrix and ε is as small as machine epsilon. In recent years, it is known that the cost of square root computation becomes almost same as that of division. Singular vectors are also computed in the same accuracy.

In 1994, Fernando and Parlett [7] considered singular value computations in terms of the pqd and dqd algorithms. The pqd algorithm does not require more computational cost per 1-step than the dqd algorithm. To accelerate the convergence a shift of origin is most important. Though the convergence speed is accelerated by introducing a shift of origin, the *shifted qd (qds) algorithm* is not always numerically stable. The qds variables may diverge to infinity by a too large shift. Fernando-Parlett proved in [7] that the *shifted differential qd (dqds) algorithm* has a wider domain of numerical stability than that of the qds algorithm. They claimed that their shifted algorithm can be used in a variety of applications, provided that all the shifts do not cause an underflow, an overflow or a devision by zero. However, in their paper we can not find how to determine such shifts. A shift which exceeds the smallest singular value may cause overflow. Error analysis and stability of the dqds algorithm was also investigated. It is to be noted that the computed singular values by the dqd algorithm are in relative error by no more than $O(m^2\varepsilon)$. Since the dqds algorithm has few roundoff error, the dqd algorithm preserves higher relative accuracy than the DK algorithm based on QR. The dqds algorithm also preserves high relative stability [7]. It has quadratic convergence or more, but it may overflow and a suitable shift-size is not known a priori. In 2000, a practical dqds algorithm is discussed by Parlett-Marques [**34**]. Though the optimum shift stratagy of the dqds algorithm has not yet been discovered, the dqds algorithm avoids numerical unstablility by programming technique on computer.

Nowadays both the DK and the dqds algorithm are useful in *Linear Algebra Package (LAPACK)* routines [**5**]. LAPACK is a freely available software package provided on the webpage [**21**] and is a library of Fortran 77 routines for solving the most commonly occurring problems in numerical linear algebra: systems of linear equations, linear least squares problems, eigenvalue problems, and singular value problems. The associated factorizations (LU, Cholesky, QR, SVD, Schur, generalized Schur) are also provided, as well as such related computations as reordering of the Schur factorizations and estimating condition numbers. Dense and banded matrices are handled, but not general sparse matrices. In many fields, a similar performance is provided for real and complex matrices, in both single and double precisions. The LAPACK library was developed at the University of Tennessee and is a de facto industry standard. Namely, LAPACK has been designed to be efficient on wide range of high-performance computers. The DK algorithm is open to the public as the routine "DBDSQ" in LAPACK. There are some LAPACK routines for computing singular values of upper bidiagonal matrices B, however only an DBDSQR routine has a good performance for computing SVD. The latest version of the dqds algorithm is also adopted as a LAPACK routine "DLASQ" for computing singular values not an SVD of B. The DLASQ routine requires only considerably less computational cost than the DBDSQR routine. In general cases, the convergence speed and numerical accuracy of DLASQ is superior than that of DBDSQR without computing singular vector. According to the detail accounts of LAPACK, the DLASQ routine is recommended for computing singular values not SVD and the DBDSQR routine should be used for SVD.

2. Integrable systems and numerical algorithms

The notion of integrability is rigidly defined for Hamilton systems. If a Hamilton system with N degree of freedom has N independent conserved quantities which are in involution, then the system of ordinary differential equations (ODEs) is said to be integrable in the sense in which the system can be linearized in terms of successive canonical transformations and be solved by quadrature. This is the main result in the Liouville-Arnold theory. Generally it is not easy to obtain explicit solutions and conserved quantities for a given nonlinear equation. For a class of integrable systems one can find explict determinantal solutions and conserved quantities with the help of Lax form and Hirota's bilinear form.

Some numerical algorithms are regarded as discrete-time dynamical systems whose soluions converge to their equilibrium points as dicrete-time goes to infinity. We can investigate the asymptotic behaviours of these dynamical systems by analyzing the explicit solutions. Such dynamical systems would be integrable systems. Moreover the discretizations of integrable systems, whose solutions converge to worthful quantites, may yeild well-known or new algorithms. Here it is important to introduce the different boundary conditions from in the case of soliton solutions. It is remarkable that the continuous Toda equation already appeared in [**36**] as a continuous limit of the qd recurrence relation. A time discretization of continous Toda equation is just the recurrence relation which appear in the qd algorithm [**14**]. It is here emphasized that the qd algorithm is shown to compute eigenvalues by using special features of the discrete Toda equation. A solution of the discrete Toda equation is written by a Hankel determinant. By an asymptotic expansion of Hankel determinant [**11**], it is shown that the solution of the discrete Toda equation converges to some limit c_k as the discrete time goes to infinity. Simultaneously, we see that the qd variable converges to c_k as the iteration number goes to infinity. A Lax form of the discrete Toda equation [**14**] plays a key role to prove that the limit c_k is eigenvalue of the given tridiagonal matrix.

There are also other various relationships between numerical algorithms and integrable systems. For example, a time-1 evolution of the continuous-time finite nonperiodic Toda equation which appears in mathematical physics is equivalent to 1-step of the QR algorithm for computing eigenvalues of a given symmetric tridiagonal matrix [**23, 42**]. There are various relationships between numerical algorithms and integrable systems. A BCH-Goppa decoding algorithm is designed by the Toda equation [**27**]. In convergence acceleration algorithms, the recurrence relations of the η-algorithm, ϵ-algorithm, the n-term of the E-algorithm are equivalent to the discrete KdV equation, full-discrete potential KdV equation and the solution of discrete hungry Lotka-Volterra equation, respectively (see [**26, 31, 45**]). The recurrence relation of the arithmetric-geometric mean algorithm can be also derived from an additional formula of the theta function.

In recent development in applied integrable systems (see [**28**]), the continuous-time Toda equation also has an application to computation of the Laplace transform of a given analytic function [**28**]. Along this line of thought some new numerical algorithms are designed in [**22, 25**] by discretizing certain integrable dynamical systems except the Toda equation. Namely, a new Padé approximation algorithms is formulated by using the relativistic Toda molecule equation (see [**22**]) and the discrete Schur flow (see [**25**]).

3. A new SVD algorithm in terms of the discrete Lotka-Volterra system

A relationship between a time-1 evolution of the *continuous-time finite Lotka-Volterra (LV) system* and 1-step of the QR algorithm which appears in the Golub-Kahan algorithm was also studied in [**1, 4**]. Here the LV system originally appears in mathematical biology and is regarded as a spatial discretization of the KdV equation. Each QR iteration for the matrix exponential traces the continuous orbit of an integrable dynamical system related to the LV system. The solutions of the LV system converge to squares of singular values of given band matrices as the time goes to infinity, respectively. However, it was not clear how to design a new numerical algorithm by discretizing the integrable system. Such a discretization scheme as the Runge-Kutta method fails to derive a "proper" recurrence system, since the discrete step-size, e.g. δ, can not be taken sufficiently large. The Runge-Kutta scheme having high accuracy converges very slowly.

A time discretization of the LV system is proposed in [**13**]. A solution of the *discrete Lotka-Volterra (dLV) system* is also expressed in a Hankel determinant form. Our starting point for designing a new SVD algorithm is to show that singular values of B are computed [**44**] by using the dLV system with the fixed discrete step-size $\delta = 1$. This is proved by using an asymptotic behaviour of Hankel determinant and a Lax form of the dLV system with $\delta = 1$. One of our devices for accelerating the convergence speed is to introduce the dLV system with arbitrary positive constant step-size $\delta > 0$. In [**15**], the dLV system with $\delta > 0$ is also shown to be applicable to singular value computation by a similar method to the dLV system with $\delta = 1$. It is shown that the convergence speed grows as δ becomes larger. However, a numerical accuracy is deteriorated by an inappropriate choice of step-size in some case. Namely, the convergence speed and the accuracy are conflicting each other in general. A flexible choice of the step-size δ at each step is desired from viewpoints of convergence speed and numerical accuracy.

In a recent development of discrete integrable systems, a *dLV (vdLV) system with variable step-size* is found in [**12, 40**]. The vdLV system differs from the dLV system with constant step-size δ in that its discrete step-size δ can be changed at each discrete time n. It is here emphasized that an explicit solution of the vdLV system is not written by a Hankel determinant but a Casorati determinant (see [**40**]). In [**15, 44**], an asymptotic expansion of Hankel determinant [**11**] is useful to prove that the solution of the dLV system converges to the singular value. However, to the best of our knowledge, any asymptotic expansion of Casorati determinant has not been

known. Hence it seems to be difficult to apply the same method for proving convergence used in the dLV system to the vdLV system. In [**16**], we have proved by a different analysis from the dLV system that the solution of the vdLV system converges to some limit. The proof is given without using the explicit form of determinant solution of the vdLV equation. By using a Lax form of the vdLV system, the constant is shown to be a singular value of the bidiagonal matrix B. We then design a numerical algorithm named the *dLV algorithm* for computing singular value. As a result, we can perform a better singular value computation with respect to both convergence speed and numerical accuracy.

In [**15**], we also describe such several features of the dLV algorithm as a sorting property of singular values, a positivity of dLV variables and so on. A new SVD algorithm named the *integrable-SVD* (*I-SVD*) *algorithm* is designed in [**17**] which can compute not only singular values but singular value vectors. Moreover we have shown that the dLV algorithm computes singular value with higher accuracy than the zero-shift LAPACK routines. For more acceleration, a new shifted integrable (sI) algorithm is designed by introducing a shift of origin into the dLV algorithm. The sI algorithm has a shift stratagy for avoiding such a numerical instability as the qds alogorithm. In some cases, the sI algorithm is superior to the nonzero-shift LAPACK routines.

4. Outline of the thesis

The thesis is organized as follows.

In Chapter 2, we prove *a determinantal solution* of the dLV system with arbitrary positive discrete step-size δ asymptotically converges to the square of some singular value of a given rectangular matrix, where the initial value of the dLV system is uniquely determined by the entries of the matrix. Here the solution means a solution expressed by determinants. To prove this fact we use an asymptotic behaviour of the Hankel determinant solution and a Lax form. A basic property of the solution is proved which is important for designing a new stable numerical algorithm. We call this algorithm the dLV algorithm for computing all of the singular values. We discuss *positivity of solution*, dependence of the correct initial value on δ, a sorting property and an acceleration of convergence speed by enlarging δ, where positivity of solution means such a property that solution is always positive.

In Chapter 3, we apply the dLV system with variable step-size to a numerical algorithm for computing singular values. A new version of the dLV algorithm is designed, where the step-size δ is replaced to a stepwise parameter $\delta^{(n)}$. Some examples demonstrate that a better choice of the step-size gives a benefit in both convergence speed and numerical accuracy.

In Chapter 4, we consider basic propeties of the dLV algorithm for computing singular values of bidiagonal matrices. A relative error bound of singular values computed by the dLV algorithm is estimated. The bound is rather smaller than that of the DK algorithm and is the same order as that of the qd algorithm. Both forward and backward stability analyses of the

dLV algorithm are also proved. A singular value computation at desired precision is carried out in terms of the Weyl type perturbation theorem. Some numerical examples illustrate a high relative accuracy of the dLV algorithm.

In Chapter 5, we present a new algorithm, named the *shifted integrable (sI) algorithm*, with a shift of origin for computing singular values σ. A shift of origin is introduced into the recurrence relation defined by the dLV system with variable step-size. A suitable shift strategy is given so that the singular value computation becomes numerically stable. The convergence of the sI algorithm is also discussed. We draw a numerical comparison among the well-known LAPACK routines and our algorithm. Our algorithm is shown to be superior to the LAPACK routines at least in four examples.

In Chapter 6, we give conclusions of this thesis and discuss further problems.

CHAPTER 2

Discrete Lotka-Volterra algorithm and its basic properties

1. Introduction

The discrete-time Lotka-Volterra (dLV) system [**13**] has determinantal solutions and a sequence of conserved quantities which are discrete analogues of those of the well known continuous-time integrable LV system. Thus we can regard the dLV system as an integrable discretization of the original integrable LV system. Interesting features of the dLV system have been studied and clarified (see [**29, 40, 41, 46**]), however, to the best of our knowledge, the role of the LV system in numerical algorithms has not yet been fully understood. In this chapter we show that a Hankel determinant solution of the finite dLV system with $\delta > 0$, where δ is the discrete step-size, converges to the square of some singular value of a given upper bidiagonal matrix $B^{(0)}$. Here a suitable initial value for the dLV system is determined by the entries of $B^{(0)}$. The proof is given by using an asymptotic behaviour of a Hankel determinant associated with a single meromorphic function [**11**]. The convergence of the qd algorithm is shown by using only an asymptotic expansion of Hankel determinat. Though the qd variable is expressed by a Hankel determinat, the solution of the dLV system is written by two types. Hence it is difficult to discuss the convergence of the solution of the dLV system by the same mannar as in the qd algorithm. It is necessary to introduce the relationship of two types of Hankel determinants.

The first purpose of this chapter is to prove a extended convergence theorem for the solution of the dLV system by starting from an alternative expression of the determinantal solution. Hankel determinants of two types appear. To describe the asymptotic behaviour of $H_{k,0}^{(n)}$ and $H_{k,1}^{(n)}$ a pair of meromorphic functions is needed which are mutually related by a linear recurrence relation. By using the asymptotics of the Hankel determinants the determinantal solution is shown to converge as discrete time $n \to \infty$ to the square of some singular value of $B^{(0)}$ for any positive δ. All of the singular values are computed in this way. The *dLV algorithm* for computing singular values is then presented.

The second purpose is to show the following basic properties of solution of the dLV system.

(i) The determinantal solution holds positive if the initial value and δ are positive, then guarantees a numerical stability of the dLV algorithm.
(ii) The correct initial value strongly depends on the discrete step-size δ.
(iii) The dLV algorithm has a sorting property, i.e., the resulting singular values are ordered according to magnitude.

(iv) The convergence to singular values is accelerated and the convergence speed tends monotonically to a limit as the discrete step-size δ increases.

These basic properties will be very important to design the dLV algorithm practically.

In §2, a determinantal solution of the dLV system with arbitrary positive δ which is characterized by two meromorphic functions is discussed. We give a new proof of the convergence of solution to some limits. In §3, these limits are shown to be the squares of singular values of a given upper bidiagonal matrix.

The basic properties (i)-(iv) are proved in §4. In particular, a close relationship between convergence speed and discrete step-size δ is described explicitly. Some numerical experiments are also presented.

2. Convergence of the determinantal solution

Let us begin with the continuous-time finite LV system

$$\begin{aligned} &\frac{du_k(t)}{dt} = u_k(t)(u_{k+1}(t) - u_{k-1}(t)), \quad k = 1, 2, \ldots, 2m-1, \\ &u_0(t) = 0, \quad u_{2m}(t) = 0, \quad t \geq 0. \end{aligned} \tag{2.1}$$

Chu [**1**] showed that a solution of (2.1) converges to the square of some singular value of a given upper bidiagonal matrix or 0 with the help of the asymptotic behaviour of solution of the finite Toda equation [**23**]. Here the initial data $\{u_{2k-1}(0), u_{2k}(0)\}$ corresponds to the entry of the bidiagonal matrix of $B^{(0)}$. Deift-Demmel-Li-Tomei [**4**] discussed a Hamiltonian structure of (2.1) and its meaning in the singuar value decomposition. However, it has not been clear how to discretize (2.1) for the purpose of designing an actual algorithm for singular value computation. We remark that (2.1) is an integrable system having the determinantal solution

$$u_{2k-1}(t) = \frac{H_{k,1}(t)H_{k-1,0}(t)}{H_{k,0}(t)H_{k-1,1}(t)}, \quad u_{2k}(t) = \frac{H_{k+1,0}(t)H_{k-1,1}(t)}{H_{k,1}(t)H_{k,0}(t)}, \tag{2.2}$$

$$H_{k,j}(t) \equiv \begin{vmatrix} a_j & a_{j+1} & \cdots & a_{j+k-1} \\ a_{j+1} & a_{j+2} & \cdots & a_{j+k} \\ \vdots & \vdots & & \vdots \\ a_{j+k-1} & a_{j+k} & \cdots & a_{j+2k-2} \end{vmatrix}(t), \quad k = 1, 2, \ldots, m,$$

$$H_{-1,j}(t) \equiv 0, \quad H_{0,j}(t) \equiv 1, \quad H_{m+1,j}(t) = 0, \quad j = 0, 1, \tag{2.3}$$

$$\frac{da_\ell(t)}{dt} = a_{\ell+1}(t), \quad \ell = 0, 1, \ldots. \tag{2.4}$$

See [**2**] for the proof of (2.4). Existense of determinantal solutions gives us a useful information for the discretization problem of integrable systems.

Several types of the dLV systems are known which have determinantal solutions. They are the infinite [**13**], semi-infinite [**40**] and finite dLV systems. In this chapter we consider the

following finite dLV system with arbitrary positive constant $\delta > 0$

$$u_k^{(n+1)}(1+\delta u_{k-1}^{(n+1)}) = u_k^{(n)}(1+\delta u_{k+1}^{(n)}), \quad k = 1, 2, \ldots, 2m-1, \tag{2.5}$$
$$u_0^{(n)} \equiv 0, \quad u_{2m}^{(n)} \equiv 0, \quad n = 0, 1, \ldots,$$

where $u_k^{(n)}$ denotes the value of u_k at discrete time $t = n\delta$. Since the dLV system (2.5) is expressed as

$$u_k^{(n+1)} - u_k^{(n)} = \delta\left(u_k^{(n)}u_{k+1}^{(n)} - u_k^{(n+1)}u_{k-1}^{(n+1)}\right),$$

it goes to the continuous-time LV system (2.1) as $\delta \to 0$ providing $t = n\delta$. Existence of the following determinantal solution is one of the reasons why we say (2.5) an integrable discretization of the continuous-time integrable LV system (2.1),

$$u_{2k-1}^{(n)} = \frac{H_{k,1}^{(n)}H_{k-1,0}^{(n+1)}}{H_{k,0}^{(n)}H_{k-1,1}^{(n+1)}}, \quad u_{2k}^{(n)} = \frac{H_{k+1,0}^{(n)}H_{k-1,1}^{(n+1)}}{H_{k,1}^{(n)}H_{k,0}^{(n+1)}}, \tag{2.6}$$

$$H_{k,j}^{(n)} \equiv \begin{vmatrix} a_j^{(n)} & a_j^{(n+1)} & \cdots & a_j^{(n+k-1)} \\ a_j^{(n+1)} & a_j^{(n+2)} & \cdots & a_j^{(n+k)} \\ \vdots & \vdots & & \vdots \\ a_j^{(n+k-1)} & a_j^{(n+k)} & \cdots & a_j^{(n+2k-2)} \end{vmatrix}, \quad k = 1, 2, \ldots, m, \ n = 0, 1, \ldots,$$

$$H_{-1,j}^{(n)} \equiv 0, \quad H_{0,j}^{(n)} \equiv 1, \quad H_{m+1,j}^{(n)} = 0, \quad j = 0, 1, \tag{2.7}$$

$$a_\ell^{(n+1)} - a_\ell^{(n)} = \delta a_{\ell+1}^{(n)}, \quad \ell = 0, 1, 2, \ldots. \tag{2.8}$$

The proof is given by using Plücker relation

$$\delta H_{k,1}^{(n)}H_{k-1,0}^{(n+1)} = H_{k-1,1}^{(n)}H_{k,0}^{(n+1)} - H_{k,0}^{(n)}H_{k-1,1}^{(n+1)} \tag{2.9}$$

and Jacobi's determinant identity

$$\delta H_{k+1,0}^{(n)}H_{k-1,1}^{(n+1)} = H_{k,0}^{(n)}H_{k,1}^{(n+1)} - H_{k,1}^{(n)}H_{k,0}^{(n+1)} \tag{2.10}$$

with the help of the linear recurrence relation (2.8). Note that (2.8), the key equation, is a simple discretization of the linear differential equation (2.4).

Let us start by introducing two functions $f_0(z)$ and $f_1(z)$ which are analytic at $z = 0$ and meromorphic in the disk $D \equiv \{z; |z| < d\}$ having the power series expansions

$$f_0(z) = \sum_{n=0}^{\infty} a_0^{(n)} z^n, \quad f_1(z) = \sum_{n=0}^{\infty} a_1^{(n)} z^n \tag{2.11}$$

at $z = 0$ and have such poles $\{z_{k,0}\}$ and $\{z_{k,1}\}$ in D that $0 < |z_{1,0}| < |z_{2,0}| < \cdots < d$ and $0 < |z_{1,1}| < |z_{2,1}| < \cdots < d$, respectively. Hankel determinants of two types, $H_{k,0}^{(n)}$ and $H_{k,1}^{(n)}$ appear in (2.6). Let the Hankel determinants $H_{k,j}^{(n)}$ be associated with the functions $f_j(z)$, namely, the coefficients $a_j^{(n)}$ of $f_j(z)$ determine $H_{k,j}^{(n)}$, $j = 0, 1$, respectively.

We here assume that a) the coefficients of $f_0(z)$ and $f_1(z)$ are related as $a_0^{(n+1)} - a_0^{(n)} = \delta a_1^{(n)}$, b) $f_j(z)$ are such rational functions of degree m that the associated Hankel determinants satisfy $H_{m+1,j}^{(n)} = 0$. The condition a) comes from the key equation (2.8) and implies

$$f_1(z) = \frac{(1-z)f_0(z) - a_0^{(0)}}{\delta z}. \tag{2.12}$$

The condition b) guarantees (2.7). It is known ([**11**], p.603) that there is a class of rational functions of degree m satisfying $H_{m+1,j}^{(n)} = 0$. The coefficients $a_j^{(n)}$ of such rational functions determine the initial value $u_k^{(0)}$ of the dLV system for $k = 1, 2, \ldots, 2m-1$ through (2.6) and (2.7).

On the other hand, an analytical property of the Hankel determinant which is associated with a meromorphic function is known ([**11**], p.596). For each k there is a nonzero constant $c_{k,j} \neq 0$ such that, for any $\rho_{k,j}$ satisfying

$$\frac{1}{|z_{k,j}|} > \rho_{k,j} > \frac{1}{|z_{k+1,j}|},$$

the Hankel determinant $H_{k,j}^{(n)}$ has an asymptotic behaviour

$$H_{k,j}^{(n)} = c_{k,j}\left(\frac{1}{z_{1,j}z_{2,j}\cdots z_{k,j}}\right)^n \left\{1 + O\left((\rho_{k,j}|z_{k,j}|)^n\right)\right\}, \quad j = 0, 1, \tag{2.13}$$

as $n \to \infty$. Substituting (2.13) into the determinantal solution (2.6) of the dLV system and using $\epsilon \equiv \max_{k,j}(\rho_{k,j}|z_{k,j}|)$, we have the following asymptotic expansion of the determinantal solution

$$\begin{aligned} u_{2k-1}^{(n)} &= \frac{c_{k,1}c_{k-1,0}\, z_{1,1}\cdots z_{k-1,1}}{c_{k,0}c_{k-1,1}\, z_{1,0}\cdots z_{k-1,0}}\left(\frac{z_{k,0}}{z_{k,1}}\right)^n \{1 + O(\epsilon^n)\}, \\ u_{2k}^{(n)} &= \frac{c_{k+1,0}c_{k-1,1}\, z_{1,0}\cdots z_{k,0}}{c_{k,1}c_{k,0}\, z_{1,1}\cdots z_{k-1,1}}\left(\frac{z_{k,1}}{z_{k+1,0}}\right)^n \{1 + O(\epsilon^n)\}, \end{aligned} \tag{2.14}$$

as $n \to \infty$.

Since we assume that $f_j(z)$ are rational functions of degree m satisfying (2.12), the poles of the rational functions $f_0(z)$ and $f_1(z)$ are coincident each other

$$z_{k,0} = z_{k,1}. \tag{2.15}$$

From the assumption we see that the poles of $f_j(z)$ have distinct modulus and ordered as $|z_{k,j}| < |z_{k+1,j}|$. Consequently, it follows from (2.14) that

$$\begin{aligned} &\lim_{n\to\infty} u_{2k-1}^{(n)} = \frac{c_{k,1}c_{k-1,0}}{c_{k,0}c_{k-1,1}} \equiv C_k, \\ &\lim_{n\to\infty} u_{2k}^{(n)} = 0, \quad k = 1, 2, \ldots, m. \end{aligned} \tag{2.16}$$

It is to be remarked that the recurrence relation (2.5) with $\delta > 0$ guarantees

$$u_k^{(n)} > 0, \quad n = 1, 2, \ldots \tag{2.17}$$

of solution for a given positive initial value $u_k^{(0)} > 0$, $k = 1, 2, \ldots, 2m-1$. Simultaneously, the limit C_k introduced by (2.16) are positive.

It is proved in this section that

Theorem 2.1. *Let the meromorphic functions $f_j(z)$ in* (2.11) *be rational functions of degree m satisfying* (2.12). *Then the solution of the finite dLV system with any positive discrete step-size δ asymptotically converges as $n \to \infty$ to some limits; the variable $u_{2k-1}^{(n)}$ with odd suffix tends to a positive limit C_k, the variable $u_{2k}^{(n)}$ goes to* 0. *The limit C_k is independent of δ.*

The meaning of the limit C_k will be discussed in the next section.

Let us here consider the asymptotic behaviour of $\{u_{2k-1}^{(n)}, u_{2k}^{(n)}\}$ in the case where $\delta = 1$. By Plücker relation (2.9) and Jacobi's determinant identity (2.10), the solution of the dLV system is rewritten as

$$u_{2k-1}^{(n)} = \frac{H_{k-1,1}^{(n)} H_{k,0}^{(n+1)}}{H_{k,0}^{(n)} H_{k-1,1}^{(n+1)}} - 1, \quad u_{2k}^{(n)} = \frac{H_{k,0}^{(n)} H_{k,1}^{(n+1)}}{H_{k,1}^{(n)} H_{k,0}^{(n+1)}} - 1. \tag{2.18}$$

The asymptotic expansion of $\{u_{2k-1}^{(n)}, u_{2k}^{(n)}\}$ in (2.18) is also given by

$$\begin{aligned} u_{2k-1}^{(n)} &= \frac{z_{1,1} z_{2,1} \cdots z_{k-1,1}}{z_{1,0} z_{2,0} \cdots z_{k,0}} \{1 + O(\epsilon^n)\} - 1, \\ u_{2k}^{(n)} &= \frac{z_{1,0} z_{2,0} \cdots z_{k,0}}{z_{1,1} z_{2,1} \cdots z_{k,1}} \{1 + O(\epsilon^n)\} - 1, \end{aligned} \tag{2.19}$$

as $n \to \infty$. This implies that $\{u_{2k-1}^{(n)}, u_{2k}^{(n)}\}$ converges to some limit as $n \to \infty$. Simultaneously, in (2.14), it is obvious that $|z_{k,0}/z_{k,1}| \leq 1$. Suppose that $z_{i-1,0} = z_{i-1,1}$ for $i = 1, 2, \ldots, k$. Then from (2.19) we derive

$$\lim_{n\to\infty} u_{2k-1}^{(n)} = \frac{1}{z_{k,0}} - 1, \quad \lim_{n\to\infty} u_{2k}^{(n)} = \frac{z_{k,0}}{z_{k,1}} - 1. \tag{2.20}$$

Note here that $1/z_{k,0} - 1$ is some positive limit. If $|z_{k,0}/z_{k,1}| < 1$, we have $|\lim_{n\to\infty} u_{2k}^{(n)} + 1| < 1$, i.e. $\lim_{n\to\infty} u_{2k}^{(n)} < 0$. Hence we see that $|z_{k,0}/z_{k,1}| = 1$, and then $u_{2k}^{(n)} \to 0$ as $n \to \infty$ in (2.14). Since $\lim_{n\to\infty} u_{2k}^{(n)} = 0$ in (2.20), we have

$$z_{k,0} = z_{k,1}. \tag{2.21}$$

Inserting (2.21) into (2.20), the solution of the dLV system converges exponentially to some limits as

$$\lim_{n\to\infty} u_{2k-1}^{(n)} = \frac{1}{z_{k,0}} - 1, \quad \lim_{n\to\infty} u_{2k}^{(n)} = 0. \tag{2.22}$$

Since $1/z_{1,0} > 1/z_{2,0} > \cdots > 1/z_{m,0}$, it follows that

Theorem 2.2. *The solution $\{u_{2k-1}^{(n)}, u_{2k}^{(n)}\}$ of the dLV system with $\delta = 1$ converges to some limit $\{1/z_{k,0} - 1, 0\}$ and $u_1^{(n)} > u_3^{(n)} > \cdots > u_{2k-1}^{(n)}$ as $n \to \infty$.*

3. Discrete Lotka-Volterra algorithm for computing singular values

Let us define new variables

$$e_k^{(n)} = \delta u_{2k-1}^{(n)} u_{2k}^{(n)}, \quad k = 1, 2, \ldots, m-1,$$
$$q_k^{(n)} = \frac{1}{\delta}\left(1 + \delta u_{2k-2}^{(n)}\right)\left(1 + \delta u_{2k-1}^{(n)}\right), \quad k = 1, 2, \ldots, m. \tag{2.23}$$

Then the dLV system (2.5) is transformed to the discrete-time finite Toda equation with the unit discrete step-size (see [**28**]), or equivalently, the recurrence relation of the qd algorithm (see [**10, 11, 35, 38**])

$$q_k^{(n+1)} e_k^{(n+1)} = q_{k+1}^{(n)} e_k^{(n)}, \quad q_k^{(n+1)} + e_{k-1}^{(n+1)} = q_k^{(n)} + e_k^{(n)},$$
$$e_0^{(n)} = 0, \quad e_m^{(n)} = 0, \quad n = 0, 1, \ldots \tag{2.24}$$

and vice versa. This type of transformation from one integrable system to another is sometimes called the Miura transformation. Let us introduce the matrices

$$Y^{(n)} \equiv L^{(n)} R^{(n)} - \frac{1}{\delta} I,$$
$$L^{(n)} \equiv \begin{pmatrix} q_1^{(n)} & & & 0 \\ 1 & q_2^{(n)} & & \\ & \ddots & \ddots & \\ & & 1 & q_m^{(n)} \end{pmatrix}, \quad R^{(n)} \equiv \begin{pmatrix} 1 & e_1^{(n)} & & \\ & 1 & \ddots & \\ & & \ddots & e_{m-1}^{(n)} \\ 0 & & & 1 \end{pmatrix}. \tag{2.25}$$

Then the Lax representation $L^{(n+1)}R^{(n+1)} = R^{(n)}L^{(n)}$ of the discrete Toda equation (2.24) gives rise to

$$Y^{(n+1)} R^{(n)} = R^{(n)} Y^{(n)}, \quad n = 0, 1, \ldots. \tag{2.26}$$

It follows from (2.16) that

$$\lim_{n\to\infty} q_k^{(n)} = (1 + \delta C_k)/\delta, \quad \lim_{n\to\infty} e_k^{(n)} = 0. \tag{2.27}$$

Therefore we have

$$\lim_{n\to\infty} Y^{(n)} = \begin{pmatrix} C_1 & & & 0 \\ 1 & C_2 & & \\ & \ddots & \ddots & \\ & & 1 & C_m \end{pmatrix}. \tag{2.28}$$

We see that C_k are the eigenvalues of the matrix $Y^{(0)} = L^{(0)}R^{(0)} - \delta^{-1}I$, since (2.26) implimes that the eigenvalues of $Y^{(0)}$ are invariant in n. This corresponds to the known fact [**10**] that $q_k^{(n)}$ converges to the eigenvalues of $L^{(0)}R^{(0)}$.

Write the tridiagonal nonsymmetric Lax matrix $Y^{(n)}$ as

$$Y^{(n)} \equiv \begin{pmatrix} w_1^{(n)} & w_1^{(n)}w_2^{(n)} & & & \\ 1 & w_2^{(n)}+w_3^{(n)} & w_3^{(n)}w_4^{(n)} & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & w_{2m-4}^{(n)}+w_{2m-3}^{(n)} & w_{2m-3}^{(n)}w_{2m-2}^{(n)} \\ & & & 1 & w_{2m-2}^{(n)}+w_{2m-1}^{(n)} \end{pmatrix},$$

$$w_k^{(n)} \equiv u_k^{(n)}(1+\delta u_{k-1}^{(n)}). \tag{2.29}$$

New variables $w_k^{(n)}$ are useful to determine a correct initial value of the dLV system in the next section. Obviously, $\lim_{n\to\infty} w_{2k-1}^{(n)} = C_k$ and $\lim_{n\to\infty} w_{2k}^{(n)} = 0$. By using the diagonal matrix

$$G^{(n)} \equiv \mathrm{diag}\left(g_{1,1}^{(n)}, \cdots, g_{m-1,m-1}^{(n)}, 1\right), \quad g_{k,k}^{(n)} \equiv \prod_{j=k}^{m-1} \sqrt{w_{2j-1}^{(n)} w_{2j}^{(n)}}, \tag{2.30}$$

we introduce a new Lax matrix

$$Y_S^{(n)} \equiv (G^{(n)})^{-1} Y^{(n)} G^{(n)} \tag{2.31}$$

which is tridiagonal and symmetric. The Lax representation (2.26) is then

$$Y_S^{(n+1)}(G^{(n+1)})^{-1}R^{(n)}G^{(n)} = (G^{(n+1)})^{-1}R^{(n)}G^{(n)}Y_S^{(n)}. \tag{2.32}$$

We note that, as $n \to \infty$, $Y_S^{(n)}$ tends asymptotically to a diagonal matrix with the eigenvalues of $Y_S^{(0)}$ on the diagonal, namely,

$$\lim_{n\to\infty} Y_S^{(n)} = \mathrm{diag}\,(C_1, C_2, \cdots, C_m), \tag{2.33}$$

where C_k are eigenvalues of $Y_S^{(0)}$. Since $Y_S^{(n)}$ is symmetric and positive definite, it admits a Cholesky decomposition

$$Y_S^{(n)} = (B^{(n)})^\top B^{(n)},$$

$$B^{(n)} \equiv \begin{pmatrix} \sqrt{w_1^{(n)}} & \sqrt{w_2^{(n)}} & & \\ & \sqrt{w_3^{(n)}} & \ddots & \\ & & \ddots & \sqrt{w_{2m-2}^{(n)}} \\ 0 & & & \sqrt{w_{2m-1}^{(n)}} \end{pmatrix}. \tag{2.34}$$

Hence the square roots $\sqrt{C_k}$ are singular values of the bidiagonal matrix $B^{(0)}$. It is concluded that

Theorem 2.3. *Let $w_k^{(0)} = b_k^2$, where b_k are nonzero entries of the $m \times m$ bidiagonal matrix $B^{(0)}$. Then the solution $u_{2k-1}^{(n)}$ of the finite dLV system with arbitrary $\delta > 0$ converges to the square of the singular value $\sigma_k \equiv \sqrt{C_k}$ of $B^{(0)}$.*

Such a class of bidiagonal matrices as

$$B^{(0)} \equiv \begin{pmatrix} b_1 & b_2 & & \\ & b_3 & \ddots & \\ & & \ddots & b_{2m-2} \\ 0 & & & b_{2m-1} \end{pmatrix}, \quad b_k \neq 0 \tag{2.35}$$

appears in the final stage of the well-known Golub-Kahan (GK) algorithm (see [**8**]) which is the standard algorithm for computing singular values of given rectangular matrices. Here the GK algorithm is a combination of the Householder transformation and the QR algorithm for the tridiagonal symmetric eigenvalue computation. The procedure of the GK algorithm is as follows. A general $m \times \ell$ rectangular matrix A, such that $m \leq \ell$, can be converted to a matrix of the form $(B^{(0)}\ O)$ by the Householder transformation as

$$U^{\top} A V = \left(B^{(0)}\ O\right),$$

where U and V are suitable orthogonal matrices and $B^{(0)}$ is such an $m \times m$ upper bidiagonal matrix as (2.30), and O is the $m \times (\ell - m)$ zero matrix (see [**8**]). The singular values of $B^{(0)}$ are congruent with those of A. Each eigenvalue of the tridiagonal matrix $(B^{(0)})^{\top} B^{(0)}$ computed by the QR algorithm gives the square of some singular value. The condition $b_k \neq 0$ implies that the singular values of A are positive and distinct. If some of b_k is zero, we can reduce the size m of the initial matrix so that every entry is not equal to zero by a deflation procedure. Singular values of any $\ell \times m$ rectangular matrix are also given by a similar way.

It is shown here that a combination of the Householder transformation and the dLV system (2.5) is also useful for computing singular values of A. Let us call this new algorithm the dLV algorithm.

4. Basic properties of the dLV algorithm

The dLV system and the dLV algorithm have the following remarkable properties.

First, we show that the determinantal solution (2.6) is always positive. As is pointed out in (2.17) the variable $u_k^{(n)}$, in n, holds positive if the initial values satisfy $u_k^{(0)} > 0$, $k = 1, 2, \ldots, 2m-1$. We can actually derive a positive sequence $\{u_k^{(0)}\}$ from any given nonzero sequence $\{b_k\}$ by

$$\begin{aligned} u_{2k-1}^{(0)} &= \frac{b_{2k-1}^2}{1 + \delta u_{2k-2}^{(0)}}, \quad k = 1, 2, \ldots, m, \\ u_{2k}^{(0)} &= \frac{b_{2k}^2}{1 + \delta u_{2k-1}^{(0)}}, \quad k = 1, 2, \ldots, m-1, \\ u_0^{(0)} &= 0, \quad u_{2m}^{(0)} = 0. \end{aligned} \tag{2.36}$$

Eq. (2.36) comes from (2.29) by setting $w_k^{(0)} = b_k^2$. Thus it is shown that

Proposition 2.4. *Singular values of $B^{(0)}$ are computed by the dLV algorithm in a numerically stable way, if $u_k^{(0)} > 0$ and $\delta > 0$.*

This property stands in contrast to the qd algorithm (see [**11**], p.613), where the recurrence relation (2.24) becomes unstable when $q_k^{(n+1)} \approx 0$. Therefore the qd algorithm was supplanted the QR algorithm in matrix eigenvalue computation [**33**]. While the dLV system is of great significance in singular value computation, though it is directly related to the qd algorithm by the Miura type transformation (2.23).

Secondly, we give comment on a revision of initial value. Eq. (2.36) also shows how to choose an initial value of the dLV algorithm for computing accurate singular values. If we set initial value as $u_{2k-1}^{(0)} = b_{2k-1}^2$ and $u_{2k}^{(0)} = b_{2k}^2$ instead of (2.36), then $u_{2k-1}^{(n)}$ does not converge to C_k. We should take the initial value as in (2.36) for any δ. The correct initial value, surprisingly, depends on the discrete step-size δ. In the limit $\delta \to 0$ the correct initial value $u_{2k-1}^{(0)}$ and $u_{2k}^{(0)}$ given by (2.36) goes to b_{2k-1}^2 and b_{2k}^2, respectively. This fact reminds us of the pathbreaking work by Chu [**1**] who showed convergence of the solution $u_{2k-1}(t)$ of the continuous-time LV system (2.1) with the initial value $u_{2k-1}(0) = b_{2k-1}^2$ and $u_{2k}(0) = b_{2k}^2$ to the squares of singular values. The basic idea in [**1**] is the asymptotics of solution of the finite Toda equation originally studied by Moser [**23**]. Here the continuous LV system is related to the Toda equation by a Miura type transformation.

Next a sorting property of the dLV algorithm is discussed. Moser [**23**] showed that the finite Toda particles are asymptotically free. By using a direct connection between the Toda equation and the LV system [**1**] it can be proved from Moser's result [**23**] that $\lim_{t\to\infty} u_{2k-1}(t) > \lim_{t\to\infty} u_{2k+1}(t)$ where $u_k(t)$ is a solution of the continuous-time LV system. This fact implies that

Proposition 2.5. *The singular values $\sigma_k = \sqrt{C_k}$ computed by the dLV algorithm are ordered according to magnitude*

$$\sigma_1 > \sigma_2 > \cdots > \sigma_k > \cdots > \sigma_m. \tag{2.37}$$

In other words, the dLV algorithm has the following sorting property

$$u_1^{(N)} > u_3^{(N)} > \cdots > u_{2k-1}^{(N)} > \cdots > u_{2m-1}^{(N)} \tag{2.38}$$

for sufficiently large N for any initial value given by (2.36). This proof is also given by combining Theorem 2.1 with Theorem 2.2.

Finally, we consider acceleration of convergence by enlarging discrete step-size δ. Since discrete time $n\delta$, for some n, becomes large as discrete step-size δ grows, we can accelerate the convergence speed. Let $u_k^{(n')}$, $n' = 0, 1, \ldots$, and $u_k^{(n)}$, $n = 0, 1, \ldots$, be solutions of the dLV system, starting from the same initial value, with discrete step-size $\delta' = \xi\delta$ and δ, respectively, for some constant $\xi > 1$. Then we see from the asymptotic expansion (2.14) that $u_k^{(n')} = u_k^{(\ell n)}$ converges faster than $u_k^{(n)}$ as $n', n \to \infty$.

More precisely we can make a clear relationship between a convergence speed and the value of δ. It is known ([**11**], p.616) that $q_k^{(n)}$ in the qd algorithm (2.24) converges to $1/z_{k,0}$, the inverse of the k-th pole $z_{k,0}$ of the meromorphic function $f_0(z)$ in (2.11). The convergence (2.27) with (2.37) says

$$z_{k,0} = \frac{\delta}{1 + \delta\sigma_k^2}, \tag{2.39}$$

where σ_k^2 is the square of the k-th largest singular value of $B^{(0)}$. The pole depends on δ. Note that $u_{2k}^{(n)} > 0$ and $\lim_{n\to\infty} u_{2k}^{(n)} = 0$. Let $u_{2j}^{(N)}$ be the largest of $\{u_{2k}^{(N)}\}$, $k = 1, 2 \ldots, m-1$, for sufficiently large N. Then we can regard $u_{2j}^{(n)}$ as the important variable which converges at the slowest. When $u_{2j}^{(N)}$ becomes $O(10^{-M})$ for some number $M > 0$, we stop the iteration of the dLV algorithm at $n = N$. This brings us a useful stopping criterion for a desired accuracy.

The convergence speed under consideration crucially depends on the ratio $z_{j,1}/z_{j+1,0}$ which appears in the asymptotic expansion of $u_{2j}^{(n)}$ as $n \to \infty$. See the second formula of (2.14). The ratio $z_{j,1}/z_{j+1,0}$ is given by the maximum of the ratios $\{z_{k,1}/z_{k+1,0}\}$, $k = 1, 2, \ldots, m-1$,

$$\frac{z_{j,1}}{z_{j+1,0}} = \max_{k=1,\ldots,m-1} \frac{z_{k,1}}{z_{k+1,0}}.$$

Using (2.15) and (2.39) we have

$$\begin{aligned} \frac{z_{j,1}}{z_{j+1,0}} &= \max_{k=1,\ldots,m-1} \frac{\sigma_{k+1}^2 + 1/\delta}{\sigma_k^2 + 1/\delta} \\ &< 1. \end{aligned}$$

It is shown that

Proposition 2.6. *The ratio $z_{j,1}/z_{j+1,0}$ decreases monotonically from* 1 *to*

$$\max_{k=1,\ldots,m-1} \frac{\sigma_{k+1}^2}{\sigma_k^2}, \tag{2.40}$$

as well as, the convergence is accelerated, as the discrete step-size δ increases from 0 *to* ∞.

It is important to manipulate the value of δ. In numerical linear algebra this kind of acceleration has not been known. This is an advantage, for improvement, of the generalized convergence theorem proved in this chapter.

We give some numerical examples below. Let

$$B^{(0)} = \begin{pmatrix} 0.5 & 0.3 & 0 \\ 0 & 0.7 & 0.1 \\ 0 & 0 & 0.9 \end{pmatrix}.$$

Figure 2.1 describes the behaviour of solution of the dLV system with $\delta = 1.0$. The solid lines indicate $\sqrt{u_1^{(n)}}$, $\sqrt{u_3^{(n)}}$ and $\sqrt{u_5^{(n)}}$ which converges to singular values. The dotted lines

correspond to $\sqrt{u_2^{(n)}}$ and $\sqrt{u_4^{(n)}}$ which tend to 0. Figure 2.2 shows an accelerated convergence of the solution, where $\delta = 10$. We see in Figures 2.1 and 2.2 that the initial value $\{\sqrt{u_k^{(0)}}\}$ depends on δ and is different from the given b_k.

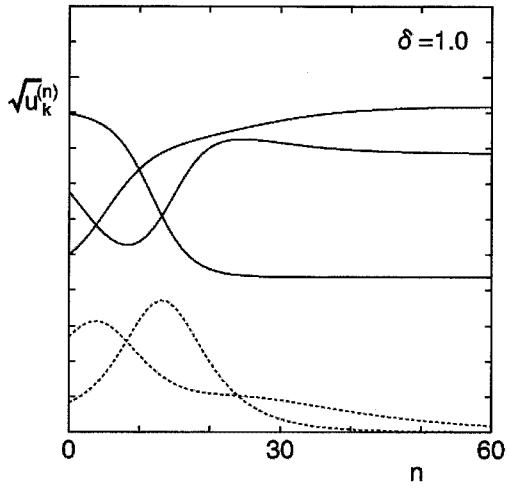

FIGURE 2.1. A graph of iteration number in the dLV algorithm (x-axis) and the square root of $u_k^{(n)}$ for $k = 1, 2, \ldots, 5$ (y-axis). The solid and dotted lines describe the behaviors of square root of $u_{2k-1}^{(n)}$, $k = 1, 2, 3$ and $u_{2k}^{(n)}$, $k = 1, 2$ from $n = 0$ to $n = 30$, respectively when $\delta = 1.0$.

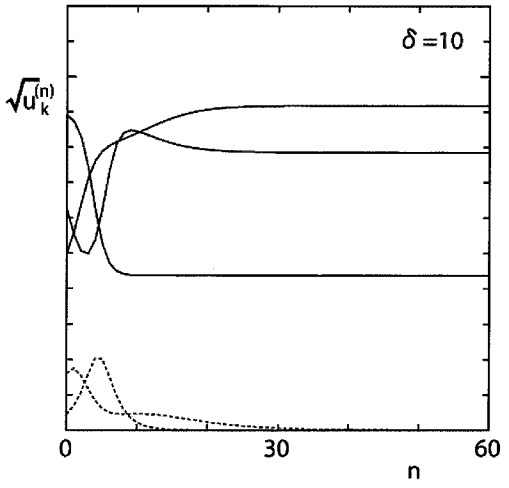

FIGURE 2.2. A graph of iteration number in the dLV algorithm (x-axis) and the square root of $u_k^{(n)}$ for $k = 1, 2, \ldots, 5$ (y-axis). The solid and dotted lines describe the behaviors of square root of $u_{2k-1}^{(n)}$, $k = 1, 2, 3$ and $u_{2k}^{(n)}$, $k = 1, 2$ from $n = 0$ to $n = 30$, respectively when $\delta = 10$.

5. Conclusion remarks

In this chapter we have proved that the Hankel determinant solution $u_{2k-1}^{(n)}$ of the dLV system converges as $n \to \infty$ to the square of the k-th largest singular value σ_k of a given bidiagonal matrix (theorems 2.1 and 2.3). And $u_{2k}^{(n)}$ tends to 0. The notions of determinantal solutions, Lax representations, Miura transformations and integrable discretizations, which have been developed in the theory of integrable systems, play a crucial role in the proof of theorems. Especially, the discretization of the linear evolution equation (2.4) to (2.8) is the key to the convergence theorem. This is because (2.8) not only characterizes the determinantal solution (2.6) but allows us arbitrary positive parameter δ and gives rise to the important relation (2.12) which enables us to obtain (2.16) through (2.14) and (2.15).

Furthermore several properties of the solution of the dLV system are discussed. For a suitable positive initial value and any positive δ, a positivity of determinantal solution is proved, which guarantees a numerical stability of the dLV algorithm (Proposition 2.4). We see in (2.36) that the correct initial value depends on discrete step-size δ. The singular values computed by the dLV system are ordered according to magnitude. Namely, the dLV algorithm has a sorting property (Proposition 2.5). As the discrete step-size δ increases from 0 to ∞, the convergence speed is accelerated to a constant determined by ratio of singular values (Proposition 2.6). A stopping criterion is also obtained by using $u_{2k}^{(n)}$. These basic properties of the dLV system, especially by a parameter δ, will see practical applications to singular value computation.

CHAPTER 3

An improvement of the discrete Lotka-Volterra algorithm

1. Introduction

Our starting point in this chapter is the observation that singular values of B are computed by using the *dLV (cdLV) system with arbitrary positive constant step-size* $\delta > 0$. Moreover, in Chapter 4, a new SVD algorithm named *integrable SVD (I-SVD) algorithm* will be discussed which can compute not only singular values but singular vectors. One of our devices for accelerating convergence speed is to enlarge δ. It is shown in Chapter 2 that convergence speed grows as δ becomes larger. However, numerical accuracy is deteriorated by an inappropriate choice of step-size in some case. Namely, convergence speed and numerical accuracy are conflicting each other in general. Though a flexible choice of the step-size δ is desired from viewpoints of convergence speed and numerical accuracy, it has not been studied how to adjust the step-size δ of the dLV system at each step.

In recent development of discrete integrable systems, a *dLV (vdLV) system with variable step-size* was also found in [**12, 40**]. It is here emphasized that an explicit solution of the vdLV system is not written by a Hankel determinant but a Casorati determinant (see [**40**]). In Chapter 2, an asymptotic expansion of Hankel determinant [**11**] is useful to prove that the solution of the cdLV system converges to the singular value. However, to the best of our knowledge, any asymptotic expansion of Casorati determinant has not been known. Hence it seems to be difficult to apply the same method of proof used in the cdLV system to the vdLV system.

In this chapter we prove by a different analysis from the cdLV system that the solution of the vdLV system converges to some limit. The proof is given without using the explicit form of determinant solution of the vdLV equation. Next we show the limit is a singular value of the bidiagonal matrix B. We then see that the vdLV system is applicable to singular value computation. A part of the I-SVD algorithm in Chapter 4 is also modified by introducing a flexible choice of the step-size δ at each step. As a result, we can perform a better singular value computation with respect to both convergence speed and numerical accuracy.

This chapter is organized as follows. In §2, it is shown that the singular values of the bidiagonal matrix B are invariant under the time evolution of the vdLV system under a suitable condition. The proof is given by using the fact that the vdLV system takes the form of a similarity transformation of a matrix. In §3, we prove convergence of solution of the vdLV system. For this purpose, it is useful to introduce the asymptotic analysis of solution of the cdLV system

in Chapter 2. Moreover, by using a relationship of the vdLV variables to the cdLV variables, it is proved that the solution of the vdLV system converge to some limit as time variable n goes to infinity. In §4, we describe two behaviours of the cdLV and the vdLV variables as n increases. Simultaneously, through some numerical examples, we demonstrate the following. A flexible choice of the step-size at each step is useful for the efficient singular value computation with respect to both convergence speed and numerical accuracy.

2. Time evolution of the vdLV system

In this section, we consider time evolution from n to $n+1$ of the finite vdLV system

$$\begin{aligned} &u_k^{(n+1)}(1+\delta^{(n+1)}u_{k-1}^{(n+1)}) = u_k^{(n)}(1+\delta^{(n)}u_{k+1}^{(n)}), \quad k=1,2,\ldots,2m-1, \\ &u_0^{(n)} \equiv 0, \quad u_{2m}^{(n)} \equiv 0, \quad 0<\delta^{(n)}<M, \quad n=0,1,\ldots, \end{aligned} \tag{3.1}$$

where $u_k^{(n)}$ and $\delta^{(n)}$ denote the value of u_k and δ, respectively, at discrete time $t=\sum_{i=0}^{n-1}\delta^{(i)}$ and M is some positive constant. If $u_k^{(0)} \geq 0$, then $u_k^{(n)} \geq 0$. The vdLV system was found in [**12, 40**] as a time discretization of the finite LV system (2.1). Namely, (3.1) goes to the LV system as every $\delta^{(n)}$ goes to zero. Let $\bar{u}_k^{(n)}$ denote the cdLV variables to distinguish the vdLV variables. The usual cdLV system, corresponds to (2.5),

$$\bar{u}_k^{(n+1)}(1+\delta\bar{u}_{k-1}^{(n+1)}) = \bar{u}_k^{(n)}(1+\delta\bar{u}_{k+1}^{(n)}) \tag{3.2}$$

is derived from (3.1) by fixing the discrete step-size $\delta^{(n)}$ at a positive constant δ.

We slightly generalize the discussion in Chapter 2. It is important to note that the vdLV variables $u_k^{(n)}$ satisfy the following matrix form:

$$\begin{aligned} &L^{(n+1)}R^{(n+1)} = R^{(n)}L^{(n)} + \left(\frac{1}{\delta^{(n+1)}} - \frac{1}{\delta^{(n)}}\right)I, \\ &L^{(n)} \equiv \begin{pmatrix} J_1^{(n)} & & & 0 \\ 1 & J_2^{(n)} & & \\ & \ddots & \ddots & \\ & & 1 & J_m^{(n)} \end{pmatrix}, \quad R^{(n)} \equiv \begin{pmatrix} 1 & V_1^{(n)} & & \\ & 1 & \ddots & \\ & & \ddots & V_{m-1}^{(n)} \\ 0 & & & 1 \end{pmatrix}, \\ &J_k^{(n)} \equiv \frac{1}{\delta^{(n)}}\left(1+\delta^{(n)}u_{2k-2}^{(n)}\right)\left(1+\delta^{(n)}u_{2k-1}^{(n)}\right), \quad V_k \equiv \delta^{(n)}u_{2k-1}^{(n)}u_{2k}^{(n)}, \end{aligned} \tag{3.3}$$

where I is the $m \times m$ unit matrix. We have the same matrix form as in Chapter 2 when $\delta^{(n)} = \delta$ for all n.

Let us begin our analysis by introducing new nonegative variables $w_k^{(n)}$ defined as

$$w_k^{(n)} = u_k^{(n)}(1+\delta^{(n)}u_{k-1}^{(n)}) \tag{3.4}$$

and a tridiagonal matrix $Y^{(n)}$

$$Y^{(n)} = L^{(n)}R^{(n)} - \frac{1}{\delta^{(n)}}I.$$

It is obvious that $Y^{(n)}$ is written as

$$Y^{(n)} = \begin{pmatrix} w_1^{(n)} & w_1^{(n)}w_2^{(n)} & & \\ 1 & w_2^{(n)} + w_3^{(n)} & \ddots & \\ & \ddots & \ddots & w_{2m-3}^{(n)}w_{2m-2}^{(n)} \\ & & 1 & w_{2m-2}^{(n)} + w_{2m-1}^{(n)} \end{pmatrix}.$$

We derive from (3.3)

$$Y^{(n+1)} = R^{(n)}Y^{(n)}(R^{(n)})^{-1}. \tag{3.5}$$

It is not hard to see $w_k^{(n)} > 0$ providing $u_k^{(0)} > 0$ and $\delta^{(n)} > 0$ for $k = 1, 2, \cdots, 2m-1$. Thus $R^{(n)}$ is nonsingular for any n. This similarity transformation (3.5) implies that the eigenvalues of $Y^{(n)}$ are invariant under the evolution from n to $n+1$ of the vdLV system. By using a diagonal matrix $G^{(n)}$, symmetrization of $Y^{(n)}$ is given as

$$\begin{aligned} Y_S^{(n)} &= (G^{(n)})^{-1}Y^{(n)}G^{(n)} \\ &= \begin{pmatrix} w_1^{(n)} & \sqrt{w_1^{(n)}w_2^{(n)}} & & \\ \sqrt{w_1^{(n)}w_2^{(n)}} & w_2^{(n)} + w_3^{(n)} & \ddots & \\ & \ddots & \ddots & \sqrt{w_{2m-3}^{(n)}w_{2m-2}^{(n)}} \\ & & \sqrt{w_{2m-3}^{(n)}w_{2m-2}^{(n)}} & w_{2m-2}^{(n)} + w_{2m-1}^{(n)} \end{pmatrix}, \\ G^{(n)} &\equiv \mathrm{diag}\left(\prod_{j=1}^{m-1} \sqrt{w_{2j-1}^{(n)}w_{2j}^{(n)}}, \prod_{j=2}^{m-1} \sqrt{w_{2j-1}^{(n)}w_{2j}^{(n)}}, \cdots, \sqrt{w_{2m-3}^{(n)}w_{2m-2}^{(n)}}, 1 \right). \end{aligned} \tag{3.6}$$

Note that $G^{(n)}$ is nonsingular for any n and $(Y_S^{(n)}) = \prod_{j=1}^{m} w_{2j-1}^{(n)}$. From (3.5) and (3.6), we have the following proposition with respect to the time evolution from n to $n+1$ of the vdLV system (3.1).

Lemma 3.1. *The vdLV system takes the form of similarity transformation*

$$Y_S^{(n+1)} = Q^{(n)}Y_S^{(n)}(Q^{(n)})^{-1}, \quad Q^{(n)} \equiv (G^{(n+1)})^{-1}R^{(n)}G^{(n)} \tag{3.7}$$

of the positive definite matrix $Y_S^{(n)}$, which implies that the eigenvalues of $Y_S^{(n)}$ are invariant under the time evolution from n to $n+1$, for all n.

It is significant to emphasize from (3.7) that choice of $\delta^{(n)}$ at each n may be made independently of the eigenvalues of $Y_S^{(n)}$. This is because the eigenvalues of $Y_S^{(n)}$ are identically equal to those of $Y^{(0)}$, i.e., the eigenvalues of $Y_S^{(n)}$ do not depend on $\delta^{(0)}, \delta^{(1)}, \cdots, \delta^{(n)}$. Note here that the

Cholesky decomposition of $Y_S^{(n)}$ is given as

$$Y_S^{(n)} = (B^{(n)})^\top B^{(n)}, \tag{3.8}$$

$$B^{(n)} \equiv \begin{pmatrix} \sqrt{w_1^{(n)}} & \sqrt{w_2^{(n)}} & & \\ & \sqrt{w_3^{(n)}} & \ddots & \\ & & \ddots & \sqrt{w_{2m-2}^{(n)}} \\ 0 & & & \sqrt{w_{2m-1}^{(n)}} \end{pmatrix}. \tag{3.9}$$

Therefore the singular values of $B^{(n)}$ are equal to the positive square roots of the eigenvalues of $Y_S^{(n)}$. Then the following proposition for the singular values of $B^{(n)}$ is derived by relating the Cholesky decomposition (3.8) to Lemma 3.1.

Proposition 3.2. *The singular values of the upper bidiagonal matrix $B^{(n)}$ are invariant under the time evolution from n to $n+1$ of the vdLV system.*

The above discussion in this section is also equivalent to that in Chapter 2, when $\delta^{(n)} = \delta$ for all n. In what follows we assume that $B^{(0)}$ has distinct singular values such that

$$\sigma_1(B^{(0)}) > \sigma_2(B^{(0)}) > \cdots > \sigma_m(B^{(0)}).$$

3. Convergence to singular values

In this section, we consider two cases where time evolution from 0 to some N is performed by the vdLV system (3.1) and by the cdLV system (3.2), respectively. Especially, in this section, we denote $Y_S^{(n)}$ and $B^{(n)}$ with $\overline{Y}_S^{(n)}$ and $\overline{B}^{(n)}$, respectively, when $\delta^{(n)} = \delta$ for all n.

It is shown in Chapter 2 that the determinantal solution $\overline{u}_k^{(n)}$ of the cdLV system (3.2) converges to some limits $\overline{c}_1, \overline{c}_2 \cdots, \overline{c}_m, 0$ as $n \to \infty$ as follows,

$$\lim_{n\to\infty} \overline{u}_{2k-1}^{(n)} = \overline{c}_k, \quad \lim_{n\to\infty} \overline{u}_{2k}^{(n)} = 0, \tag{3.10}$$

where $\overline{c}_1 > \overline{c}_2 > \cdots > \overline{c}_m > 0$. Simultaneously, it is proved that $\overline{c}_k$ are eigenvalues of $\overline{Y}_S^{(n)}$ and $\sqrt{\overline{c}_k}$ are singular values of $\overline{B}^{(0)}$. The proof of (3.10) is given by an asymptotic expansion of the determinantal solution $\overline{u}_k^{(n)}$ as $n \to \infty$. However, it seems difficult to apply the same analysis to the vdLV system. In this section, we investigate the asymptotic behaviour as $n \to \infty$ of the vdLV variables $u_k^{(n)}$ by using the property (3.10) of the cdLV system instead of the asymptotic expansion of the solution.

Let us begin our analysis by considering trace$(Y_S^{(n)})$. As shown in §2, the eigenvalues of $Y_S^{(n)}$ are independent from the choice of $\delta^{(0)}, \delta^{(1)}, \cdots, \delta^{(n)}$. Namely,

$$\lambda(Y_S^{(n)}) = \lambda(\overline{Y}_S^{(n)}), \tag{3.11}$$

where $\lambda(Y_S^{(n)})$ and $\lambda(\overline{Y}_S^{(n)})$ denote the eigenvalues of $Y_S^{(n)}$ and $\overline{Y}_S^{(n)}$, respectively. Note that $\lambda(\overline{Y}_S^{(n)}) = \overline{c}_k$. Then we see from (3.11) that $\lambda(Y_S^{(n)}) = \overline{c}_k$. In general, the sum of all diagonal entries coincides with that of all eigenvalues. Moreover it is obvious from (3.6) that $\mathrm{trace}(Y_S^{(n)}) = \sum_{k=1}^{2m-1} w_k^{(n)}$. Hence we have

$$\sum_{k=1}^{2m-1} w_k^{(n)} = \sum_{k=1}^{m} \overline{c}_k. \tag{3.12}$$

Namely, $\sum_{k=1}^{2m-1} w_k^{(n)}$ are invariant in n. This fact is useful to prove the following lemma for analyzing the behaviour of the vdLV variables $u_k^{(n)}$ as $n \to \infty$.

Lemma 3.3. *Suppose the initial data $u_k^{(0)}$ is such that*

$$u_k^{(0)} > 0, \quad k = 1, 2, \ldots, 2m-1. \tag{3.13}$$

Then $u_k^{(n)}$, $n = 0, 1, \ldots$, satisfy

$$0 < u_k^{(n)} < M_1, \quad k = 1, 2, \ldots, 2m-1. \tag{3.14}$$

for some positive constant M_1.

Proof. It is obvious from (3.1) that $u_k^{(n)} > 0$, for all n, under initial data (3.13). From (3.4) and (3.12), we see that $w_k^{(n)} > 0$ and $0 < w_1^{(n)} + w_2^{(n)} + \cdots + w_{2m-1}^{(n)} < M_2$ for some constant M_2. Hence $0 < w_k^{(n)} < M_2$ for all n. By using (3.4), we have (3.14). □

With the help of Lemma 3.3, the behaviour of $u_k^{(n)}$ as $n \to \infty$ with initial data (3.13) is described by the following Proposition.

Proposition 3.4. *If $u_k^{(0)}$ satisfy initial data* (3.13)*, then*

$$\lim_{n\to\infty} u_{2k-1}^{(n)} = \overline{c}_k, \ \lim_{n\to\infty} u_{2k}^{(n)} = 0. \tag{3.15}$$

Proof. Let $k = 1$ in the vdLV system (3.1), then we have $u_1^{(N+1)} = u_1^{(0)} \prod_{n=0}^{N} (1 + \delta^{(n)} u_2^{(n)})$ for some N which implies that $u_1^{(0)} \le u_1^{(1)} \le \cdots u_1^{(n)} \le \cdots$. From Lemma 3.3, it is obvious that $0 < u_1^{(n)} < M_1$ for all n. Since $u_1^{(n)}$, $n = 0, 1, \ldots$, is monotonically increasing, $u_1^{(n)}$ converges to some positive limit c_1 as $n \to \infty$. Simultaneously, $\prod_{n=0}^{\infty} (1 + \delta^{(n)} u_2^{(n)})$ converges to some positive limit p_1.

Suppose that $\prod_{n=1}^{\infty} (1 + \delta^{(n)} u_{2k-2}^{(n)})$ converges to some positive limit p_{k-1}. Let

$$v_k^{(0)} = u_k^{(0)} (1 + \delta^{(0)} u_{k+1}^{(0)})$$

and $v_k^{(0)} > 0$. Then, by using (3.1) and $0 < \delta^{(n)} < M$, we see that $(v_{2k-1}^{(0)} / p_{k-1}) \prod_{n=1}^{N} (1 + \delta^{(n)} u_{2k}^{(n)})$ converges to $u_{2k-1}^{(N+1)}$ as $N \to \infty$. Hence it follows from (3.14) that $0 < \prod_{n=0}^{\infty} (1 + \delta^{(n)} u_{2k}^{(n)}) < M_3$ for some constant M_3. It is also obvious that $\prod_{n=1}^{N} (1 + \delta^{(n)} u_{2k}^{(n)})$, $N = 1, 2, \ldots$, is monotonically

increasing. Therefore it follows that $\prod_{n=1}^{\infty}(1+\delta^{(n)}u_{2k}^{(n)}) = p_k$. Simultaneously, we see that $\lim_{n\to\infty} u_{2k-1}^{(n)} = v_{2k-1}^{(0)}p_k/p_{k-1} > 0$, namely,

$$\lim_{n\to\infty} u_{2k-1}^{(n)} = c_k \tag{3.16}$$

where c_k is some positive limit.

Note here that $\sum_{n=0}^{\infty}\delta^{(n)}u_{2k}^{(n)}$ converges to some limit $s_k > 0$ if and only if $\prod_{n=1}^{\infty}(1+\delta^{(n)}u_{2k}^{(n)}) = p_k$ for $\delta^{(n)}u_{2k}^{(n)} > 0, n = 0, 1, \ldots$. Moreover $\lim_{n\to\infty}\delta^{(n)}u_{2k}^{(n)} = 0$ for any positive bounded sequence $\delta^{(n)}$, if $\sum_{n=0}^{\infty}\delta^{(n)}u_{2k}^{(n)} = s_k$. Therefore it follows that

$$\lim_{n\to\infty} u_{2k}^{(n)} = 0.$$

The vdLV system (3.1) also leads to

$$\lim_{n\to\infty} u_{2k-2}^{(n+1)} = v_{2k-2}^{(0)}\prod_{n=1}^{\infty}(1-\delta^{(n)}r_k^{(n)})$$

where $\delta^{(n)}r_{k-1}^{(n)} < 1$ and $r_{k-1}^{(n)} \equiv (u_{2k-3}^{(n)} - u_{2k-1}^{(n)})/(1+\delta^{(n)}u_{2k-3}^{(n)})$. If $c_{k-1} = c_k$ i.e. $\lim_{n\to\infty} r_{k-1}^{(n)} = 0$, then $\lim_{n\to\infty} u_{2k-2}^{(n+1)} \neq 0$. Since $\lim_{n\to\infty} u_{2k-2}^{(n+1)} = 0$, we see that $c_{k-1} \neq c_k$. If $c_{k-1} < c_k$, then $\lim_{n\to\infty} r_{k-1}^{(n)} < 0$ and $\lim_{n\to\infty} u_{2k-2}^{(n+1)} = \infty$. Otherwise $\lim_{n\to\infty} r_{k-1}^{(n)} > 0$ and $\lim_{n\to\infty} u_{2k-2}^{(n+1)} = 0$. Hence we have $c_1 > c_2 > \cdots > c_m$. This sorting property is the same as that of the cdLV system (3.2) as $n \to \infty$. It is to be remarked that $\lim_{N\to\infty}\prod_{n=1}^{N}(1-\delta^{(n)}r_k^{(n)}) = 0$.

Note here that $\lim_{n\to\infty} Y_S^{(n)} = \mathrm{diag}(c_1, c_2, \cdots, c_m)$. This implies that c_k is the eigenvalue of $Y_S^{(n)}$. By using (3.11), it turns out that c_k coincides with one of the eigenvalues $\bar{c}_1, \bar{c}_2, \cdots, \bar{c}_m$ of $\bar{Y}_S^{(n)}$. Since $c_1 > c_2 > \cdots > c_m$ and $\bar{c}_1 > \bar{c}_2 > \cdots > \bar{c}_m$, it follows that

$$c_k = \bar{c}_k. \tag{3.17}$$

Consequently we have (3.15). □

Combining Proposition 3.4 with Proposition 3.2, we derive the following theorem for the singular value of $B^{(0)}$, or equivalently, $\bar{B}^{(0)}$.

Theorem 3.5. *The k-th singular value $\sigma_k(B^{(0)})$ of $B^{(0)}$ with nonzero diagonal and subdiagonal entries is equal to $\sqrt{c_k}$, namely,*

$$\sigma_k(B^{(0)}) = \sqrt{c_k},$$

for $k = 1, 2, \ldots, m$, where c_k is the limit of $u_{2k-1}^{(n)}$ as $n \to \infty$.

Proof. It is proved in Proposition 3.2 that the singular values of $B^{(n)}$ are invariant in n. From the asymptotic analysis in Proposition 3.4, we have $B^{(n)} \to \mathrm{diag}(\sqrt{c_1}, \sqrt{c_2}, \cdots, \sqrt{c_m})$ as $n \to \infty$. Hence it follows from $c_k = \bar{c}_k$ that $\sqrt{c_k}$ is the k-th singular value of $B^{(0)}$. □

The Cholesky decomposition (3.8) guarantees that the initial data $w_k^{(0)}$ is always nonnegative in singular value computation. We also see from (3.4) that $u_k^{(0)} \geq 0$. Let us consider the case where $u_k^{(0)} = 0$ for some k. If $u_{2k_0-1}^{(0)} = 0$ for $0 < k_0 \leq m$, we compute the eigenvalue of $(\hat{B}^{(0)})^\top \hat{B}^{(0)}$ defined by $(\hat{B}^{(0)})^\top \hat{B}^{(0)} = (B^{(0)})^\top B^{(0)} + \sigma^2 I$ where σ is some positive constant. Then all entries of $\hat{B}^{(0)}$ are nonzero positive. Note that $\lambda_k((B^{(0)})^\top B^{(0)})$ is given by $\lambda_k((\hat{B}^{(0)})^\top \hat{B}^{(0)}) - \sigma^2$. If $u_{2k_0}^{(0)} = 0$, then

$$B^{(0)} = \begin{pmatrix} B_1^{(0)} & 0 \\ 0 & B_2^{(0)} \end{pmatrix}$$

where $B_1^{(0)} \in \boldsymbol{R}^{k_0 \times k_0}$ and $B_2^{(0)} \in \boldsymbol{R}^{(m-k_0)\times(m-k_0)}$ are upper bidiagonal matrices. Hence the singular value computation of $B^{(0)}$ can be performed by computing singular values of $B_1^{(0)}$ and $B_2^{(0)}$. Therefore it is enough to discuss the initial data (3.13) in Theorem 3.5.

As is shown in Theorem 3.5, the vdLV system (3.1) is applicable to singular value computation. In the next section, we explain an advantage which the I-SVD algorithm with variable step-size has.

4. Numerical examples

First, we show some numerical results computed by using a part of the I-SVD algorithm with variable step-size. To investigate the effect of variable step-size let us take up the following simple example,

$$B_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Let us compare Case 1 with Case 2 shown in Table 3.1. Each asymptotic behaviour of $\sqrt{u_k^{(n)}}$

TABLE 3.1. Choice of the step-size $\delta^{(n)}$

	step-size $\delta^{(n)}$
Case 1	$\delta^{(0)} = 1, \delta^{(1)} = 100, \delta^{(2)} = 1, \delta^{(3)} = 100, \cdots$
Case 2	$\delta^{(n)} = 1$ for $n = 0, 1, \cdots$
Case 3	$\delta^{(n)} = 100$ for $n = 0, 1, \cdots$
Case 4	$\delta^{(0)} = 1, \cdots, \delta^{(10)} = 1, \delta^{(11)} = 100, \cdots$
Case 5	$\delta^{(0)} = 100, \cdots, \delta^{(10)} = 100, \delta^{(11)} = 1, \cdots$

and $\sqrt{w_k^{(n)}}$ is shown in Figure 3.1 and Figure 3.2 respectively. Figures 3.1 and 3.2 demonstrate that the variables $\sqrt{u_k^{(n)}}$ and $\sqrt{w_k^{(n)}}$ converge to some limits independently of the choice of $\delta^{(n)}$. Though the values of $\sqrt{u_k^{(n)}}$ vibrate, those of $\sqrt{w_k^{(n)}}$ do not. From the discussions in §2, we can regard $\sqrt{w_{2k-1}^{(n)}}$ as the singular value of B $(= B^{(0)})$, when $\sqrt{w_{2k-2}^{(n)}}$ is approximately equal to

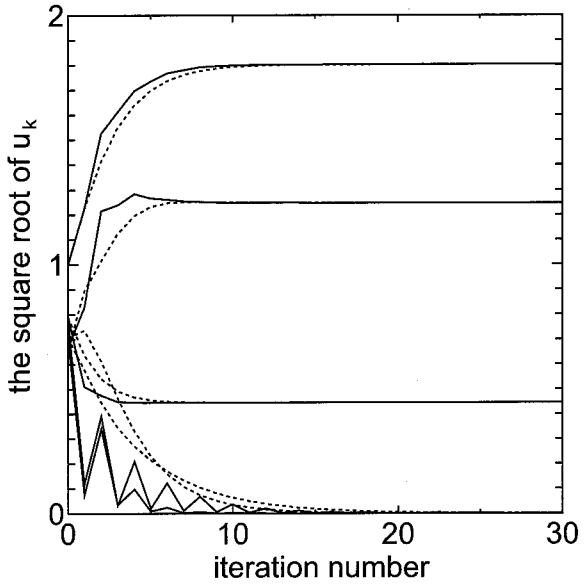

FIGURE 3.1. A graph of iteration number in a part of I-SVD algorithm (x-axis) and the square root of $u_k^{(n)}$ for $k = 1, 2, \ldots, 5$ (y-axis). The solid and dotted lines describe the behaviors of square root of $u_k^{(n)}$ from $n = 0$ to $n = 30$ in Case 1 and Case 2, respectively.

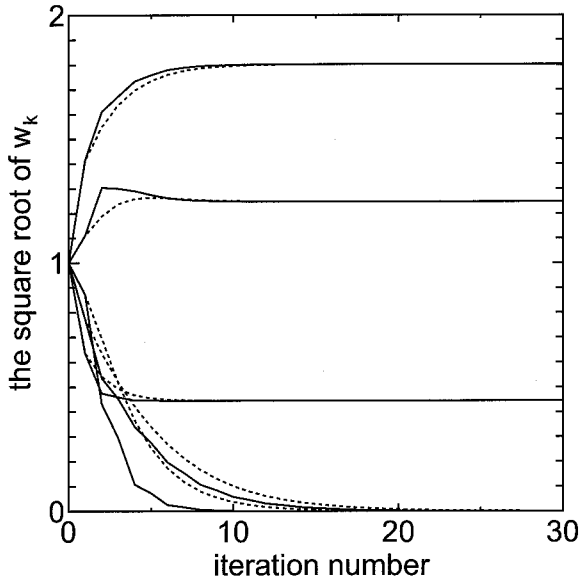

FIGURE 3.2. A graph of iteration number in a part of I-SVD algorithm (x-axis) and the square root of $w_k^{(n)}$ for $k = 1, 2, \ldots, 5$ (y-axis). The solid and dotted lines describe the behaviors of square root of $w_k^{(n)}$ from $n = 0$ to $n = 30$ in Case 1 and Case 2, respectively.

0. Thus $w_{2k-2}^{(\bar{n})} \approx 0$, for some $\bar{n}$, gives rise to a stopping criterion. In practical computation, we adopt $\sqrt{w_{2k-2}^{(n)}} < \varepsilon$ as the stopping criterion where ε is a small number. When $\sqrt{w_{2k-2}^{(n)}} < \varepsilon$, we can reduce the matrix-size m to $m-1$. This is a deflation process.

Next, we examine Cases 2-5 in Table 3.1. Figure 3.3 describes the behaviour of $\sqrt{u_2^{(n)}}$, especially in Cases 2-5. Figure 3.3 suggests some benefit derived from the flexible choice of step-size $\delta^{(n)}$. Compared Case 4 with Cases 2, 3, it turns out that convergence speed is accelated as $\delta^{(n)}$ becomes larger on the way of iterations. It is shown in Chapter 2 that the convergence speed tends to a constant determined by a ratio of singular values by enlarging δ. We also see from Case 5 of Figure 3.3 that convergence speed is reduced by decreasing the value of $\delta^{(n)}$.

However, Case 5 has an advantage with respect to computational cost since we can replace $\delta^{(n)} * u_k^{(n)}$ with $u_k^{(n)}$ from $n = 10$. Let us compare Case 5 with Cases 2,3 and 4 through the singular value computation of

$$B_2 = \begin{pmatrix} 8.4 & 7.6 & 0 \\ 0 & 4 & 0.02 \\ 0 & 0 & 0.01 \end{pmatrix},$$

where $\varepsilon \equiv 1.0 \times 10^{-16}$. Table 3.2 shows timing of deflation in Cases 2-5. For example, in

TABLE 3.2. Timing of deflation in Cases 2-5

	First deflation (matrix-size $m : 3 \rightarrow 2$)	Second deflation (computation is completed)
Case 2	$n = 30$	$n = 30$
Case 3	$n = 10$	$n = 29$
Case 4	$n = 18$	$n = 29$
Case 5	$n = 10$	$n = 29$

Case 3, it turns out from Table 3.2 we compute the singular values of 3×3 matrix for $n \leq 10$ and 2×2 for $11 \leq n \leq 29$. In consideration that $\delta^{(n)} u_k^{(n)} = u_k^{(n)}$ when $\delta^{(n)} = 1$, the operation number in Cases 2-5 is shown as Table 3.3. It is significant to note here that division needs more

TABLE 3.3. Operation number in Cases 2-5

	Additions	Multiplications	Divisions
Case 2	240	125	120
Case 3	156	239	78
Case 4	188	207	94
Case 5	156	163	78

computational cost than multiplication. Hence, from viewpoint of computational cost, Case 5 is

better than other four cases. Simultaneouly, the roundoff error in Case 5 is also small since the operation number is the small. Consequently, the flexible choice of $\delta^{(n)}$ at each step is shown to be useful for an efficient computation with respect to both convergence speed and numerical accuracy.

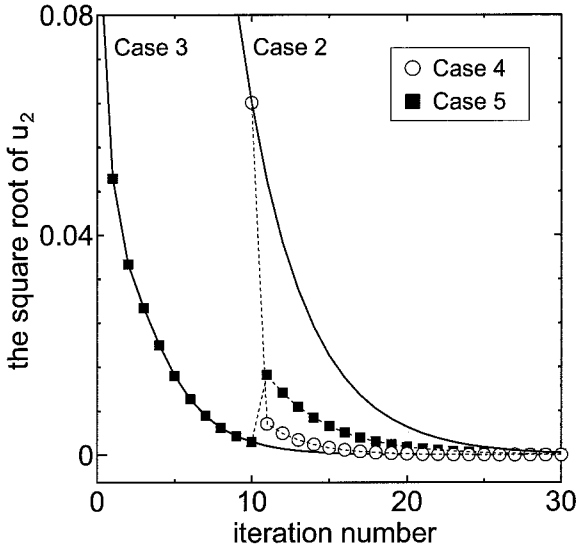

FIGURE 3.3. A graph of iteration number in a part of I-SVD algorithm (x-axis) and the square root of $u_2^{(n)}$ (y-axis). The solid lines describe the behaviors of square root of $u_2^{(n)}$ from $n = 0$ to $n = 30$ in Case 2 and Case 3. The white circle and black square marks correspond to the square root of $u_2^{(n)}$ from $n = 0$ to $n = 30$ in Case 4 and Case 5, respectively.

Although a suitable stratagy of $\delta^{(n)}$ has not been found generally, we explain how to determine $\delta^{(n)}$ by relating it to the singular values $\sigma_k(B)$, $k = 1, 2, \ldots, m$. A proper choice of the variable step-size $\delta^{(n)}$ may depend on the distribution of singular values. It is shown in Chapter 2 that the convergence speed crucially depends on the ratio of nearest singular values, i.e., the value of $\max_k(\sigma_{k+1}^2(B)/\sigma_k^2(B))$. Note that

$$\left|\frac{w_{2k}^{(N+1)}}{w_{2k}^{(N)}}\right| = \frac{u_{2k+1}^{(N)} + 1/\delta^{(N)}}{u_{2k-1}^{(N)} + 1/\delta^{(N)}}, \quad \lim_{n\to\infty} u_{2k-1}^{(n)} = \sigma_k^2(B),$$

since $w_{2k}^{(N+1)} = v_{2k}^{(0)} \prod_{n=1}^{N}((1 + \delta^{(n)}u_{2k+1}^{(n)})/(1 + \delta^{(n)}u_{2k-1}^{(n)}))$. Then we see that the convergence speed of the series $\{w_{2k}^{(n)}\}_{n=0,1,\ldots}$ grows as $\delta^{(n)}$ becomes larger. If $\max_k(\sigma_{k+1}^2(B)/\sigma_k^2(B))$ is not close to 1, a good acceleration is performed by enlarging $\delta^{(n)}$. However any control of positive $\delta^{(n)}$ hardly gives rise to acceleration when the distribution is dense. Hence $\delta^{(n)} = 1$ will be a better choice in this case with respect to the operation number and the roundoff error.

Let us consider the case where $\sigma_1(B) \approx \sigma_2(B) \approx \cdots \approx \sigma_{m-1}(B)$ and the smallest singular value $\sigma_m(B)$ is not close to $\sigma_{m-1}(B)$. Then $w_{2m-2}^{(n)}$ primarily converges to zero after performing some repeat loops. A larger step-size $\delta^{(n)}$ accelerates this convergence. It follows from Proposition 3.2 that $w_{2m-2}^{(N)} \approx 0$ and $w_{2m-1}^{(N)} \approx \sigma_m^2(B)$ for some N. Therefore we can introduce a deflation from $B \in \boldsymbol{R}^{m\times m}$ to $\tilde{B} \in \boldsymbol{R}^{(m-1)\times(m-1)}$. Since $\tilde{B}$ has dense singular values, we switch the $\delta^{(n)}$ to 1. This control of $\delta^{(n)}$ corresponds to that in Case 5. Similarly, we may adjust such $\delta^{(n)}$ as in Case 4, when $\sigma_k(B)$, $k = 1, 2, \ldots, m-2$ are sufficiently separated each other and $\sigma_{m-1}(B) \approx \sigma_m(B)$.

CHAPTER 4

On the discrete Lotka-Volterra algorithm: error analysis, stability and singular vectors

1. Introduction

A new algorithm for computing singular values named the dLV algorithm is presented in previous chapter. The dLV algorithm can be visualized by Figure 4.1 named the *dLV Table*. The dLV Table describes a rhombus-like rule and seems to be very similar to the qd Table

$$
\begin{array}{ccccccccc}
 & b_1^2 & \cdots & b_{2k-2}^2 & b_{2k-1}^2 & b_{2k}^2 & \cdots & b_{2m-1}^2 & \\
u_0^{(0)} & u_1^{(0)} & \cdots & u_{2k-2}^{(0)} & u_{2k-1}^{(0)} & u_{2k}^{(0)} & \cdots & u_{2m-1}^{(0)} & u_{2m}^{(0)} \\
u_0^{(1)} & u_1^{(1)} & \cdots & u_{2k-2}^{(1)} & u_{2k-1}^{(1)} & u_{2k}^{(1)} & \cdots & u_{2m-1}^{(1)} & u_{2m}^{(1)} \\
u_0^{(2)} & u_1^{(2)} & \cdots & u_{2k-2}^{(2)} & u_{2k-1}^{(2)} & u_{2k}^{(2)} & \cdots & u_{2m-1}^{(2)} & u_{2m}^{(2)} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & \sigma_1^2 & \cdots & 0 & \sigma_k^2 & 0 & \cdots & \sigma_m^2 & 0
\end{array}
$$

FIGURE 4.1. dLV Table

[**11**]. As is pointed out in Chapter 2, there is an intimate relationship between the dLV system (2.5) and the pqd recurrence relation (2.17). Namely, the qd variables and the LV variables are directly connected by Miura transformation (2.23). Both the pqd algorithm [**7, 32**] and the dLV algorithm, shown in the previous chapter can compute, singular values of bidiagonal matrices B without square root computation. Table 4.1 gives a comparison of the complexity of pqd, dqd and dLV iterations. See [7] for the complexity of DK and *orthogonal qd (oqd)* algorithms.

It is possible to apply the dLV algorithm to a wide class of rectangular matrices A by using the Householder transformation from A to $(B\ O)$ or $(B^\top\ O)^\top$. The dLV algorithm has the following advantages. The singular value computation is performed only by additions, multiplications and divisions, whereas it does not need any subtraction in each iteration as well as

TABLE 4.1. Complexity of pqd, dqd and dLV algorithms

	pqd	dqd	dLV
Square roots	0	0	0
Divisions	1	1	1
Multiplications	1	2	1
Additions	1	1	2
Subtractions	1	0	0
Assignments	2	3	1

square root computation. It is well known that numerical errors may be extremely large in algorithm which uses subtractions with multiplications or divisions. The dLV algorithm avoids this situation. Since the initial value given by entries of the upper bidiagonal matrix B is nonnegative, every quantity is also positive at any time (see Chapter 2). The dLV variables, keep positive, guarantee

$$1 < 1 + \delta u_{k-1}^{(n+1)}, \tag{4.1}$$

where $1 + \delta u_{k-1}^{(n+1)}$ is the denominator of the dLV recurrence relation (2.5). We can choose such a parameter δ that $1 + \delta u_{k-1}^{(n+1)} < M$ for a certain positive number M. Hence high numerical stability of the dLV algorithm may follow.

A dLV algorithm having variable step-size $\delta^{(n)}$ is given in Chapter 3. It has better convergence speed than the dLV algorithm with $\delta = 1$. The sI algorithm is also presented in Chapter 5. The speed is drastically increased. In many numerical experiments the sI algorithm is rather faster than DBDSQR (without singular vectors computation) in LAPACK. Here the shifted DK and the dqds algorithm are implemented in DBDSQR and DLASQ, respectively. Moreover, accuracy of the sI algorithm is better than these today's standard packages. Hence reliable approaches, such as error analysis and stability analysis, to basic features of the dLV and sI algorithms are worthwhile. In this chapter some basic properties of the dLV algorithm having constant step-size δ are discussed.

The first purpose of this chapter is concerning with error analysis. It is necessary to verify a high relative accuracy which results from the nonnegativity of the dLV variables. We consider errors of the dLV algorithm through the following two approaches.

The first is an estimation of relative error bound of 1-iteration of the dLV algorithm. Using a method by Demmel [**5**] and Fernando-Parlett [7] it is shown that singular values are computed by the dLV algorithm with a high relative accuracy. Namely computed singular value by the dLV algorithm is in relative error by no more than $O(m^2\varepsilon)$ which is as same order as that by the dqd algorithm, where ε is as small as machine epsilon. The dLV algorithm is more accurate than the DK algorithm. The other approach to errors is a singular value computation for a desired

accuracy by the dLV algorithm. For this end, it is useful to introduce Weyl type perturbation theorem in [**30**] suited for numerical inclusion of matrix singular values. This theorem says that errors of the singular values are estimated by the computational errors of matrices. The errors of matrices are evaluated by changing a roundoff mode. In this process, we use two types of roundoff mode defined as

(a) *Down* : Round $c \in \boldsymbol{R}$ to the largest floating point number $f \in \boldsymbol{F}$ satisfying $f \leq c$,

(b) *Up* : Round $c \in \boldsymbol{R}$ to the smallest floating point number $f \in \boldsymbol{F}$ satisfying $f \geq c$,

where $\boldsymbol{R}$ and $\boldsymbol{F}$ denote the sets of real numbers and floating point numebers, respectively. Therefore, we have rigorous error bounds for the computed singular values. It is shown that the dLV algorithm computes singular values at a high precision.

The second purpose of this chapter is concerning with stability analysis of the dLV algorithm. The method in [**5, 7**] is also applicable to prove forward and backward stability analyses of the dLV algorithm. Both forward and backward errors of 1-step are shown to be $O(m\varepsilon)$.

This chapter is organized as follows. In §2, we estimation of relative error bound of 1-step of the dLV algorithm and ensure a high relative accuracy of the algorithm. In §3, forward and backward stability analyses of the dLV algorithm are proved. In §4, we prepare a procedure for computing singular vectors in terms of the dLV algorithm. It is possible to estimate an error bound of singular values computed by the dLV algorithm. Some numerical examples for comparison of the dLV, the DK and the pqd algorithms for singular values are given in §5.

2. Error analysis for the dLV algorithm

It is shown in [**6**] that the error bound on singular values after 1-step of the DK algorithm without shift is $69m^2\varepsilon$. The error bound of the dqd algorithm is $4m\varepsilon$ [**7**] which is rather smaller than that of the DK algorithm.

An error analysis for the dLV algorithm can be done along a similar line to [**7**]. Let B be such a given upper bidiagonal matrix that $b_k \neq 0$ for $k = 1, 2, \ldots, 2m-1$ and δ be some positive constant. Set $w_k^{(0)} = b_k^2$ and $\gamma = 1/\delta$. Then the dLV algorithm for computing singular values of B is formulated. We write 1-step of the dLV algorithm by using a modification of the variable $w_k^{(n)} \equiv u_k^{(n)}(1 + \delta u_{k-1}^{(n)})$ such that $w_k^{(n)} \equiv u_k^{(n)}(\gamma + u_{k-1}^{(n)})$ as follows,

$$
\begin{aligned}
&u_k = \frac{w_k}{\gamma + u_{k-1}}, \quad k = 1, 2, \ldots, 2m-1, \\
&\widehat{w}_{k-1} = u_{k-1}(\gamma + u_k), \quad k = 2, \ldots, 2m, \\
&u_0 = 0, \quad u_{2m} = 0,
\end{aligned} \tag{4.2}
$$

where $w_k = w_k^{(n)}$ and $\widehat{w}_k = w_k^{(n+1)}$. The convergence theorem proved in Chapter 2 says $\lim_{n\to\infty} w_{2k-1}^{(n)} = \gamma\sigma_k^2$ and $\lim_{n\to\infty} w_{2k}^{(n)} = 0$ for any positive γ.

Introduce the set of $2m-1$ quantities

$$W = \{w_1, w_2, \cdots, w_{2m-1}\}. \tag{4.3}$$

Given W 1-step of the dLV algorithm in finite precision arithmetic generates output $\widehat{W} = \{\widehat{w}_1, \widehat{w}_2, \ldots, \widehat{w}_{2m-1}\}$. Let $\vec{W}$ be a set with small relative perturbation of W. Let $\check{W}$ be the output of the dLV algorithm acting on $\vec{W}$ in exact arithmetic computation. We require that $\widehat{W}$ is a set with a small relative perturbation of $\check{W}$. These sets are mutually related as Figure 4.2.

$$\begin{array}{ccc} W & \xrightarrow{\text{dLV/computed}} & \widehat{W} \\ \text{perturbation of } w \big\downarrow & & \big\uparrow \text{perturbation of } \check{w} \\ \vec{W} & \xrightarrow[\text{dLV/exact}]{} & \check{W} \end{array}$$

FIGURE 4.2. Effects of roundoff/W diagram

We estimate a relative error bound of the singular value computed by the dLV algorithm. Let the floating point computation of a basic arithmetic operation $\circ$ satisfy $fl(x \circ y) = (x \circ y)(1+\eta) = (x \circ y)/(1+\delta)$, where $|\eta| < \varepsilon$ and $|\delta| < \varepsilon$ for a given ε. This is the arithmetic model in [7]. A relationship between W quantities and $\widehat{W}$ quantities computed by the dLV algorithm is as follows.

$$u_k = \frac{w_k}{\gamma + u_{k-1}} \frac{1+\varepsilon_/}{1+\varepsilon_{k-1}}, \tag{4.4}$$

$$\begin{aligned} \widehat{w}_{k-1} &= u_{k-1}(\gamma + u_k)(1+\varepsilon_k)(1+\varepsilon_*) \\ &= u_{k-1}\left(\gamma + \frac{w_k}{\gamma + u_{k-1}} \frac{1+\varepsilon_/}{1+\varepsilon_{k-1}}\right)(1+\varepsilon_k)(1+\varepsilon_*), \end{aligned} \tag{4.5}$$

where $|\varepsilon_/| < \varepsilon$ and so on. All the ε's depend on k. We note the k-dependence of relative error arising from addition.

Let us introduce a small relative perturbation

$$\vec{w}_k = w_k(1+\vec{\varepsilon}_1)/(1+\vec{\varepsilon}_2) \tag{4.6}$$

of W to $\vec{W}$. Then exact computation by the dLV algorithm gives $\check{W}$ quantities from $\vec{W}$ as

$$\vec{u}_k = \frac{w_k}{\gamma + \vec{u}_{k-1}} \frac{1+\vec{\varepsilon}_1}{1+\vec{\varepsilon}_2}, \tag{4.7}$$

$$\begin{aligned} \check{w}_{k-1} &= \vec{u}_{k-1}(\gamma + \vec{u}_k) \\ &= \vec{u}_{k-1}\left(\gamma + \frac{w_k}{\gamma + \vec{u}_{k-1}} \frac{1+\vec{\varepsilon}_1}{1+\vec{\varepsilon}_2}\right). \end{aligned} \tag{4.8}$$

Let us set

$$\widetilde{w}_k = \check{w}_k(1+\check{\varepsilon}_1)(1+\check{\varepsilon}_2). \tag{4.9}$$

Then we have from (4.8)

$$\widetilde{w}_{k-1} = \vec{u}_{k-1}\left(\gamma + \frac{w_k}{\gamma + \vec{u}_{k-1}}\frac{1+\vec{\varepsilon}_1}{1+\vec{\varepsilon}_2}\right)(1+\check{\varepsilon}_1)(1+\check{\varepsilon}_2). \tag{4.10}$$

If we choose $\vec{\varepsilon}_1 = \varepsilon_/$, $\vec{\varepsilon}_2 = \varepsilon_{k-1}$, $\check{\varepsilon}_1 = \varepsilon_*$, $\check{\varepsilon}_2 = \varepsilon_k$, then

$$\vec{u}_k = \frac{w_k}{\gamma + \vec{u}_{k-1}}\frac{1+\varepsilon_/}{1+\varepsilon_{k-1}}. \tag{4.11}$$

This implies from (4.4) that $\vec{u}_k = u_k$. Therefore we see $\widetilde{w}_{k-1} = \widehat{w}_{k-1}$ from (4.5) and (4.10).

Theorem 4.1. *The W diagram commutes and $\vec{w}_k$ differs from w_k by 2ε at most, $\widehat{w}_k$ differs from $\check{w}_k$ by 2ε at most. The dLV algorithm with $\delta > 0$ guarantees that each computed singular value of $m \times m$ bidiagonal matrices is in error by no more than $(4m-2)\varepsilon$.*

Corollary 4.2. *If there is no roundoff error in addition of unity, each computed singular value is in error by no more than $(2m-1)\varepsilon$ for the dLV algorithm with $\delta = 1$.*

3. Backward and forward stabilities

There are two types of roundoff error analysis. One is forward error analysis and the other is backward. For a given set of data $z \equiv \{z_k\}$ let us write some exact computation on this data by $C(z) \equiv C(z_1, z_2, \cdots, z_m)$. For $\overline{z} \equiv \{\overline{z}_k\}$ we write $C(\overline{z}) \equiv C(\overline{z}_1, \overline{z}_2, \cdots, \overline{z}_m)$ similarly. Let $fl(C(z))$ denote output generated by 1-step of the algorithm considered in finite precision arithmetic. The term *forward error analysis* is to determine a forward error as

$$\|C(z) - fl(C(z))\|. \tag{4.12}$$

If the forward error of computation of $fl(C(z))$ from z is small, such a computation is said to be forward stable. *Backward error analysis* requires to find exact value $\overline{z} = \{\overline{z}_k\}$ which satisfies

$$fl(C(z_1, z_2, \cdots, z_m)) = C(\overline{z}_1, \overline{z}_2, \cdots, \overline{z}_m). \tag{4.13}$$

The difference between z and $\overline{z}$ indicates the backward error of the computation of $fl(C(z))$ from z. If the backward error is small, such a computation is said to be backward stable. See Figure 4.3.

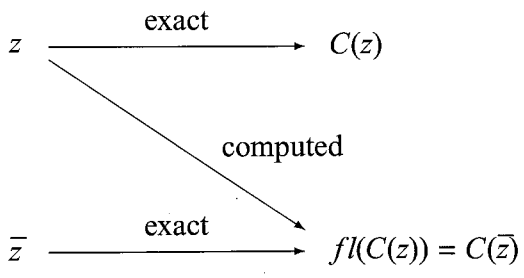

FIGURE 4.3. Forward and backward errors

Figure 4.4 shows multiple sweeps of the dLV algorithm. The mappings $W_l \to W_{l+1}$, $W_l \to \vec{W}_l$, $\vec{W}_l \to \check{W}_{l+1}$, $\check{W}_{l+1} \to W_{l+1}$ in Figure 4.4 are corresponding to $W \to \widehat{W}$, $W \to \vec{W}$, $\vec{W} \to \check{W}$, $\check{W} \to \widehat{W}$ in Figure 4.2, respectively. The actual computation proceeds as

$$W_l \to W_{l+1} \to W_{l+2} \to W_{l+3}. \tag{4.14}$$

Let us consider the computation

$$\vec{W}_l \to \vec{W}_{l+1} \to \vec{W}_{l+2} \to \vec{W}_{l+3}. \tag{4.15}$$

Then we obtain $\check{W}_{l+1}$ as an exact computing result of $\vec{W}_l$. And $\vec{W}_{l+1}$ is a floating point computing result of $\vec{W}_l$. Moreover it is shown that the difference between $\check{W}_{l+1}$ and $\vec{W}_{l+1}$ is $4(2m-1)\varepsilon$ and is small. Hence the computation of $\vec{W}_{l+1}$ from $\vec{W}_l$ is forward stable.

Similarly, we consider the computation which proceeds as

$$\check{W}_l \to \check{W}_{l+1} \to \check{W}_{l+2} \to \check{W}_{l+3}. \tag{4.16}$$

We obtain $\check{W}_{l+2}$ from $\check{W}_{l+1}$ as a floating point computing result. And $\check{W}_{l+2}$ is also an exact computing result of $\vec{W}_{l+1}$. It can be shown that the error between $\check{W}_{l+1}$ and $\vec{W}_{l+1}$ is $4(2m-1)\varepsilon$. Hence the computation of $\check{W}_{l+2}$ from $\check{W}_{l+1}$ is backward stable.

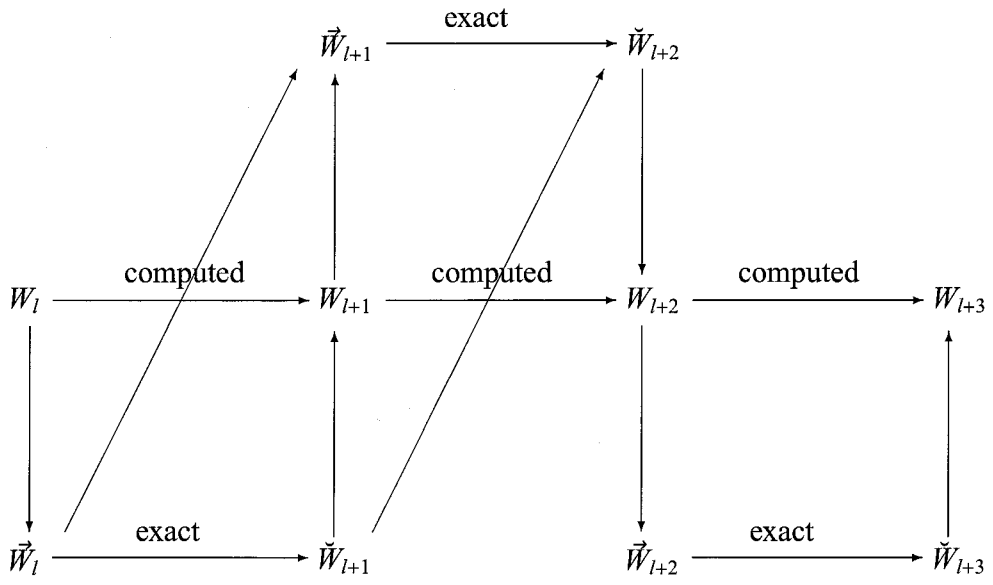

FIGURE 4.4. Effects of roundoff for multiple sweep of dLV algorithm

Let B be such an upper bidiagonal matrix as (2.35). Then no overflow and no underflow occur. Therefore it is concluded that

Theorem 4.3. *The dLV algorithm is forward and backward stable.*

4. Singular value computation for a desired accuracy

If the $m \times m$ bidiagonal matrix B is approximately decomposed, by some numerical algorithm, into

$$B = U\Sigma V^{\top}, \quad \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_m), \tag{4.17}$$

where σ_k are singular values and the column vectors of U and V are singular vectors of B, we can estimate an error bound of the computed singular values in terms of the singular vectors. For this purpose the extended Weyl type perturbation theorem by Oishi [**30**] is most useful.

Theorem 4.4. *Let B be an $m \times m$ real matrix. We assume that as a result of any numerical computation algorithm we have an $m \times m$ real diagonal matrix Σ and $m \times m$ orthogonal matrices U and V such that*

$$U\Sigma V^{\top} = B + E, \quad U^{\top}U = I + F, \quad V^{\top}V = I + G, \tag{4.18}$$

where E, F and G are matrices expressing computational errors. We assume that $\|F\|_2 < 1$ and $\|G\|_2 < 1$ so that U and V may be invertible. where $\|F\|_2$ and $\|G\|_2$ denote 2-norms of F and G, respectively. Let $\widetilde{\sigma}_k$ and σ_k be singular values of B and Σ, respectively, where $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \cdots \geq \widetilde{\sigma}_m$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m$. Then the following estimation holds:

$$|\sigma_k - \widetilde{\sigma}_k| \leq |\widetilde{\sigma}_k| \max\{\|F\|_2, \|G\|_2\} + \|E\|_2. \tag{4.19}$$

This theorem enables us to estimate error bounds of singular values from the computational errors of singular vectors computed by a numerical SVD procedure. The quantities $\|E\|_2$, $\|F\|_2$ and $\|G\|_2$ in (4.19) are evaluated by changing a roundoff mode.

In this section we formulate an SVD procedure for the bidiagonal matrix B to discuss a precision of singular values computed by the dLV algorithm. To this end a close relationship between the dLV recurrence relation (3.5) and the qd recurrence relation (2.24) is fundamental.

Let us give a brief review of the pqd algorithm for computing eigenvalues [**37, 38**]. The qd recurrence relation (2.24) has such an asymptotic behaviour [**11**] that $q_k^{(n)} \to c_k$, $e_k^{(n)} \to 0$ as $n \to \infty$ under a suitable assumption, where c_k are some nonzero limits satisfying $|c_1| > |c_2| > \cdots > |c_m| > 0$. Let us begin with the LR matrix expression of the qd recurrence relation

$$L^{(n+1)}R^{(n+1)} = R^{(n)}L^{(n)}, \tag{4.20}$$

where $L^{(n)}$ and $R^{(n)}$ are given by (2.25). Introduce a sequence of $m \times m$ tridiagonal matrices

$$T^{(n)} \equiv \begin{pmatrix} q_1^{(n)} & q_1^{(n)} e_1^{(n)} & & & \\ 1 & e_1^{(n)} + q_2^{(n)} & q_2^{(n)} e_2^{(n)} & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & q_{m-1}^{(n)} e_{m-1}^{(n)} \\ & & & 1 & e_{m-1}^{(n)} + q_m^{(n)} \end{pmatrix}$$
$$= L^{(n)} R^{(n)}, \quad n = 0, 1, \ldots. \tag{4.21}$$

Then the LR expression (4.20) takes the form of similarity transformation

$$T^{(n)} = R^{(n)^{-1}} T^{(n+1)} R^{(n)}. \tag{4.22}$$

This is a discrete self-similar flow. Note that the eigenvalues of $T^{(n)}$ are invariant under the time evolution from n to $n+1$. Let us write the eigenvalues of $T^{(n)}$ as $\lambda_k(T^{(n)})$. The iteration (4.22) yields such a decomposition of $T^{(0)}$ that

$$T^{(0)} = \left(R^{(1)} R^{(2)} \cdots R^{(n-1)}\right)^{-1} T^{(n)} R^{(1)} R^{(2)} \cdots R^{(n-1)}. \tag{4.23}$$

When $q_k^{(n)} \to c_k$ and $e_k^{(n)} \to 0$ as $n \to \infty$, $T^{(n)}$ also converges to a lower bidiagonal matrix as $n \to \infty$,

$$T^{(\infty)} = \begin{pmatrix} c_1 & & & 0 \\ 1 & c_2 & & \\ & \ddots & \ddots & \\ & & 1 & c_m \end{pmatrix}. \tag{4.24}$$

Since c_k are the eigenvalues of $T^{(\infty)}$, it follows from $\lambda_k(T^{(\infty)}) = \lambda_k(T^{(0)})$ that c_k are the eigenvalues of $T^{(0)}$. Simultaneously, $R^{(1)} R^{(2)} \cdots R^{(n-1)}$ converges to an upper triangular matrix. Therefore the qd algorithm computes real nonzero eigenvalues c_k of the tridiagonal matrix $T^{(0)}$.

An eigendecomposition of $T^{(\infty)}$ is also given in terms of the qd algorithm.

Lemma 4.5. *An eigendecomposition of the bidiagonal matrix $T^{(\infty)}$ is given by*

$$T^{(\infty)} = P \Lambda P^{-1},$$
$$P \equiv \begin{pmatrix} p_{1,1} & & & & 0 \\ p_{2,1} & p_{2,2} & & & \\ \vdots & \vdots & \ddots & & \\ p_{m-1,1} & p_{m-1,2} & \cdots & p_{m-1,m-1} & \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix}, \quad p_{i,j} \equiv \prod_{k=i+1}^{m} \left(c_j - c_k\right),$$
$$\Lambda \equiv \mathrm{diag}(c_1, c_2, \cdots, c_m). \tag{4.25}$$

The decomposition (4.23) and the eigendecomposition (4.25) lead to the following lemma for the eigendecomposition of $T^{(0)}$ in terms of the qd algorithm.

Lemma 4.6. *An eigendecomposition of the tridiagonal matrix $T^{(0)}$ is given by*

$$T^{(0)} = R_p^{(\bar{n})^{-1}} P \Lambda P^{-1} R_p^{(\bar{n})}, \quad R_p^{(\bar{n})} \equiv R^{(1)} R^{(2)} \cdots R^{(\bar{n}-1)}, \tag{4.26}$$

where $\bar{n}$ is such a stopping time that $T^{(\bar{n})} = R_p^{(\bar{n})} T^{(0)} R_p^{(\bar{n})^{-1}}$ is approximately lower bidiagonal.

Next we extend the eigendecomposition of $T^{(0)}$ to that of a class of symmetric tridiagonal matrices. Let us consider the case where the initial data of the qd algorithm is given by

$$q_k^{(0)} = b_{k,k}^2 > 0, \quad e_k^{(0)} = b_{k,k+1}^2 > 0. \tag{4.27}$$

We can symmetrize $T^{(0)}$ as

Lemma 4.7. *Symmetrization of $T^{(0)}$ by a similarity transformation is given by*

$$\begin{aligned} T_{\mathrm{s}} &= G T^{(0)} G^{-1}, \\ G &\equiv \mathrm{diag}(g_{1,1}, \ldots, g_{m-1,m-1}, 1), \quad g_{k,k} \equiv \prod_{j=k}^{m-1} \frac{1}{b_{2j-1} b_{2j}}, \end{aligned} \tag{4.28}$$

where the symmetric matrix T_{s} is written by

$$T_{\mathrm{s}} = \begin{pmatrix} b_1^2 & b_1 b_2 & & & \\ b_1 b_2 & b_2^2 + b_3^2 & b_3 b_4 & & \\ & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & & & \\ & & b_{2m-5} b_{2m-4} & b_{2m-4}^2 + b_{2m-3}^2 & b_{2m-3} b_{2m-2} \\ & & & b_{2m-3} b_{2m-2} & b_{2m-2}^2 + b_{2m-1}^2 \end{pmatrix}. \tag{4.29}$$

Proof. Let us set $G = \mathrm{diag}(g_{1,1}, g_{2,2}, \cdots, g_{m,m})$. Then the (k,k)-entries of $G T_{\mathrm{s}} G^{-1}$ are $b_{2k-2}^2 + b_{2k-1}^2$. It is obvious that $(k, k+1)$-entry and $(k+1, k)$-entry are $b_{2k-1}^2 b_{2k}^2 g_{k,k} / g_{k+1,k+1}$ and $g_{k+1,k+1} / g_{k,k}$, respectively. If $g_{k,k} = \prod_{j=k}^{m-1} 1 / b_{2j-1} b_{2j}$ and $g_{m,m} = 1$, both $(k, k+1)$-entry and $(k+1, k)$-entry become $b_{2k-1} b_{2k}$. □

Note that T_{s} is positive definite, namely, $c_k > 0$. By applying Lemma 4.7 to Lemma 4.6, we have an eigendecomposition of positive definite symmetric matrices:

Lemma 4.8. *An eigendecomposition of T_{s} is given as follows*

$$T_{\mathrm{s}} = V \Lambda V^{-1}, \quad V \equiv G R_p^{(\bar{n})^{-1}} P. \tag{4.30}$$

Let us set $V \equiv (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m)$, where $\boldsymbol{v}_j \equiv (v_{1j}, v_{2j}, \ldots, v_{mj})^\top$ and each $v_{i,j}$ represents the (i,j)-entry of V. The vector $\boldsymbol{v}_j$ is just an eigenvector for the eigenvalue $\lambda_j(T_{\mathrm{s}})$ computed by the qd algorithm. Note that $\boldsymbol{v}_i^\top \boldsymbol{v}_j = 0$ in the case where $\lambda_i(T_{\mathrm{s}}) \neq \lambda_j(T_{\mathrm{s}})$ for $i \neq j$ since T_{s} is

symmetric and $V^\top V\Lambda = \Lambda V^\top V$. Thus we can transform V to an orthogonal matrix $\bar{V} = (\bar{v}_{i,j})$ by the normalization $\boldsymbol{v}_i \to \bar{\boldsymbol{v}}_i = (\bar{v}_{i,1}, \bar{v}_{i,2}, \ldots, \bar{v}_{i,m})^\top$ such as $\bar{v}_{i,1}^2 + \bar{v}_{i,2}^2 + \cdots + \bar{v}_{i,m}^2 = 1$. The result is

$$\bar{v}_{i,j} = \frac{v_{i,j}}{\sqrt{v_{i,1}^2 + v_{i,2}^2 + \cdots + v_{i,m}^2}}, \quad j = 1, 2, \ldots, m. \tag{4.31}$$

Here each $\bar{\boldsymbol{v}}_i$ corresponds to a unit eigenvector of $\lambda_i(T_s)$. Consequently, we have an eigendecomposition of T_s by the orthogonal matrix $\bar{V}$.

Lemma 4.9. *An eigendecomposition of T_s is given by*

$$T_s = \bar{V}\Lambda\bar{V}^\top, \tag{4.32}$$

where $\bar{V}$ is an orthogonal matrix given by normalizing the columns of V.

Now we consider an SVD of such a bidiagonal matrix B as (2.35). Any positive definite symmetric matrix T_s in (4.29) admits the Cholesky decomposition of the form

$$T_s = B^\top B. \tag{4.33}$$

Since the limits c_k are simple eigenvalues of the positive definite matrix T_s, Lemma 4.9 says that the qd algorithm computes the singular values of B through

$$\sigma_k(B) = \sqrt{c_k}, \tag{4.34}$$

such that $\sigma_1 > \sigma_2 > \cdots > \sigma_m > 0$. Moreover, for some orthogonal matrix $\bar{U}$, the decompositions (4.32) and (4.33) generate

$$B^\top B = \left(\bar{U}\Lambda^{\frac{1}{2}}\bar{V}^\top\right)^\top \bar{U}\Lambda^{\frac{1}{2}}\bar{V}^\top, \quad \Lambda^{\frac{1}{2}} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_m). \tag{4.35}$$

We have

Lemma 4.10. *An SVD of B is presented as $B = \bar{U}\Sigma\bar{V}^\top$, where $\Sigma = \Lambda^{1/2}$. Here the orthogonal matrices $\bar{V}$ and $\bar{U}$ are given by normalizing column vectors of $V = GR_p^{(\bar{n})^{-1}}P$ and $\bar{U} = B\bar{V}\Sigma^{-1}$, respectively.*

The diagonal matrix Σ of singular values and the orthogonal matrix $\bar{V}$ of singular vectors are shown to be computed by the qd algorithm with the initial data (4.27). Since the convergence of the qd algorithm without shift is very slow, it needs many times of matrix product to compute $R_p^{(\bar{n})}$. Thus this SVD procedure is impractical. However it is useful to discuss a singular value computation, for a desired accuracy, by the dLV algorithm.

Let B be such a bidiagonal matrix as (2.35). An SVD in terms of the dLV algorithm is described as follows.

(i) Set a suitable discrete step-size $\delta > 0$ and the initial data by (2.36).

(ii) By using the dLV algorithm the singular values σ_k of B are computed as

$$\sigma_k^2 = u_{2k-1}^{(\bar{n})}, \quad \text{for such } \bar{n} \text{ that } \max_{j=1,\ldots,m-1} u_{2j}^{(\bar{n})} \leq 1.0 \times 10^{-\alpha} \tag{4.36}$$

for some positive integer α, where the second condition is a stopping criterion.

(iii) Compute the product of upper bidiagonal matrices $R_p^{(\bar{n})} = R^{(0)}R^{(1)} \cdots R^{(\bar{n}-1)}$ and its inverse, where each $R^{(j)}$ is given by replacing $e_k^{(j)}$ with $\delta u_{2k-1}^{(j)} u_{2k}^{(j)}$ in (4.20).

(iv) Prepare the diagonal matrix G and a lower triangular matrix P with $c_k = \sigma_k^2 > 0$.

(v) Through the nonsingular matrix $V = G{R_p^{(\bar{n})}}^{-1}P$, the orthogonal matrices $\bar{V}$ and $\bar{U} = B\bar{V}\bar{\Sigma}^{-1}$ such that

$$B = \bar{U}\bar{\Sigma}\bar{V}^{\top}, \tag{4.37}$$

are obtained.

We call this the *integrable SVD (I-SVD) algorithm.*

It becomes possible to estimate error bounds of singular values σ_k computed in Steps (i)-(ii) through the SVD procedure (iii)-(v). An example of the singular value computation at desired precision is given in the next section.

5. Numerical examples

In this section we first give some numerical examples for comparison of the dLV algorithm with the DK and the pqd algorithms without shift. For the DK algorithm we take up DBDSQR of LAPACK code, where both the shift and the singular vector computation routines are excluded. For the pqd and the dLV algorithms, the Demmel-Kahan QR routine of DBDSQR is replaced by the pqd without shift and the dLV routines, respectively. The same stopping criterion is adopted as that of DBDSQR. We fix the parameter as $\delta = 1$ for the dLV algorithm, for simplicity.

Here we consider 100×100 and 500×500 matrices of four types in Table 4.2, where $\hat{\sigma}_k$ are the verified correct values. We show the singular values $\hat{\sigma}_k$ of the 100×100 matrices

TABLE 4.2. Four cases of upper bidiagonal matrices

	Diagonal b_{2k-1}	Subdiagonal b_{2k}	Distribution of $\hat{\sigma}_k$	Minimal $\hat{\sigma}_m$
Case 1 : B_1	2.001	2	sufficiently separated	nonzero
Case 2 : B_2	1	10	somewhat separated	almost zero
Case 3 : B_3	0.001	$\begin{cases} 2 & (k=1) \\ 1 & (\text{otherwise}) \end{cases}$	dense (except for $\hat{\sigma}_m$)	nonzero
Case 4 : B_4	2	0.001	dense (except for $\hat{\sigma}_m$)	almost zero

B_i, $i = 1, 2, 3, 4$ in Table 4.3.

TABLE 4.3. Singular values in four cases of 100×100 matrices

	Distribution of 100 singular values
Case 1	$\hat{\sigma}_1 = 4.000511306\cdots, \hat{\sigma}_2 = 3.999045346\cdots,$ $\ldots, \hat{\sigma}_{99} = 0.094010676\cdots, \hat{\sigma}_{100} = 0.031906725\cdots$
Case 2	$\hat{\sigma}_1 = 10.99955222\cdots, \hat{\sigma}_2 = 10.99820922\cdots,$ $\ldots, \hat{\sigma}_{99} = 9.000549469\cdots, \hat{\sigma}_{100} = 0.000000000\cdots$
Case 3	$\hat{\sigma}_1 = 2.001999014\cdots, \hat{\sigma}_2 = 2.001996057\cdots,$ $\ldots, \hat{\sigma}_{99} = 1.998000987\cdots, \hat{\sigma}_{100} = 0.999999833\cdots$
Case 4	$\hat{\sigma}_1 = 2.000999506\cdots, \hat{\sigma}_2 = 2.000998027\cdots,$ $\ldots, \hat{\sigma}_{99} = 1.999000493\cdots, \hat{\sigma}_{100} = 0.000000000\cdots$

Table 4.4 gives computational time of the DK, the pqd and the dLV algorithms with $\delta = 1$ for these matrices. The numerical experimentation was carried out on our computer with CPU: Pentium III 933MHz, RAM: 512MB and every quantities were computed in the double precision. When singular values are dense, convergence becomes very slow in these algorithms. Since the computation of the minimal singular value σ_{100} or σ_{500} is completed at the early stage of iterations, there is no definite influence of the existence of almost zero-singular value. Table 4.4 also suggests a scalability of the dLV algorithm.

TABLE 4.4. Computational time of the DK, the pqd and the dLV algorithms (sec.)

	100×100			500×500		
	DK	pqd	dLV	DK	pqd	dLV
Case 1	0.20	0.05	0.07	6.92	1.37	2.35
Case 2	0.93	0.20	0.30	33.44	6.44	9.86
Case 3	79.40	16.33	30.11	2523.43	525.30	979.82
Case 4	149.07	31.52	58.20	4915.28	1013.60	1902.86

Next we discuss accuracy of singular values computed by the DK, the pqd and the dLV algorithms for B_i, $i = 1, 2, 3, 4$, where every B_i is 100×100.

Figure 4.5 describes relative errors $|\sigma_k - \hat{\sigma}_k|/\hat{\sigma}_k$ of the computed singular values σ_k of B_1. To see a difference in accuracy among the algorithms, we rearrange these relative errors from small to large. Then the resulting Figure 4.6 shows that the relative errors computed by the dLV algorithm are slightly smaller than those by the others.

Relative errors for the matrix B_2 with $m = 100$ are given in Figure 4.7. Since B_2 has an almost zero-singular value $\hat{\sigma}_{100}$, the relative error for σ_{100} is replaced with the absolute error $|\sigma_{100} - \hat{\sigma}_{100}|$ in Figure 4.7. The relative errors of the dLV algorithm are more or less smaller than the others.

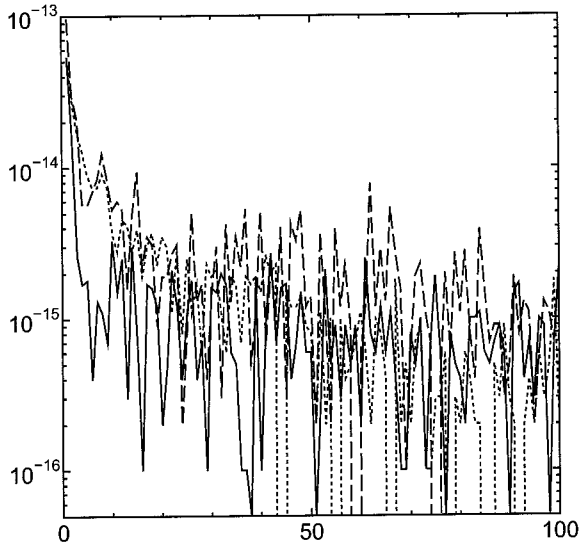

FIGURE 4.5. A graph of the suffix k for ordering singular values σ_k according to magnitude (x-axis) and relative errors in computed singular values of B_1 by the DK, the pqd and the dLV algorithms (y-axis). The dashed, dotted and solid lines are given by the DK, the pqd and the dLV algorithms, respectively.

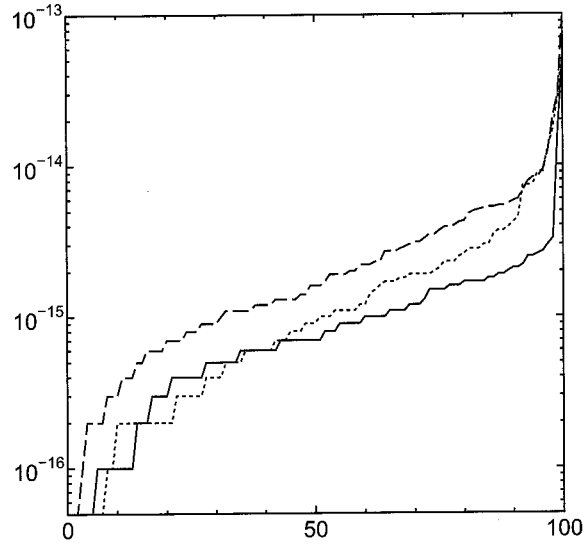

FIGURE 4.6. A graph of rearranged relative errors in computed singular values σ_k of B_1 by the DK, the pqd and the dLV algorithms from small to large. The dashed, dotted and solid lines are given by the DK, the pqd and the dLV algorithms, respectively.

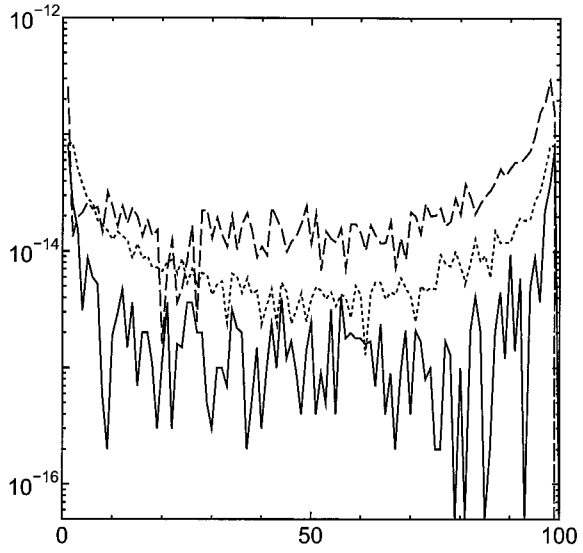

FIGURE 4.7. A graph of the suffix k for ordering singular values σ_k according to magnitude (x-axis) and relative errors in computed singular values of B_2 by the DK, the pqd and the dLV algorithms (y-axis). The dashed, dotted and solid lines are given by the DK, the pqd and the dLV algorithms, respectively.

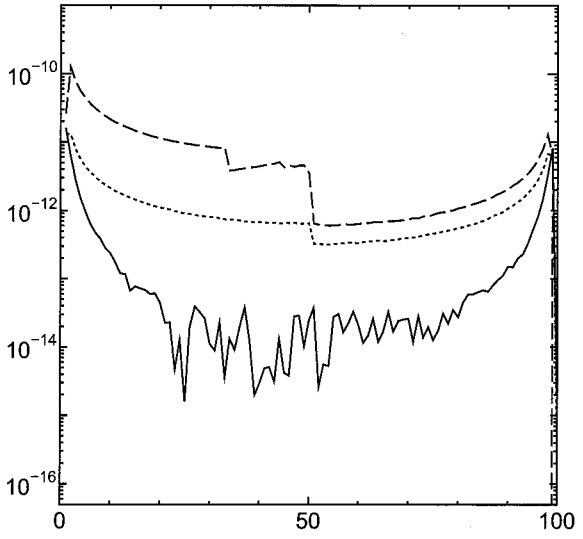

FIGURE 4.8. A graph of the suffix k for ordering singular values σ_k according to magnitude (x-axis) and relative errors in computed singular values of B_3 by the DK, the pqd and the dLV algorithms (y-axis). The dashed, dotted and solid lines are given by the DK, the pqd and the dLV algorithms, respectively.

Relative errors for the matrices B_3 and B_4 with $m = 100$ are described in Figure 4.8 and 4.9, respectively. The absolute error $|\sigma_{100} - \hat{\sigma}_{100}|$ is plotted in place of the relative error for σ_{100} in Figure 4.9. The quantity $u_{199}^{(n)}$ in Case 4 converges fast to $\sigma_{100}^2 \approx 0$ in the dLV algorithm. The dLV algorithm is the most accurate of all.

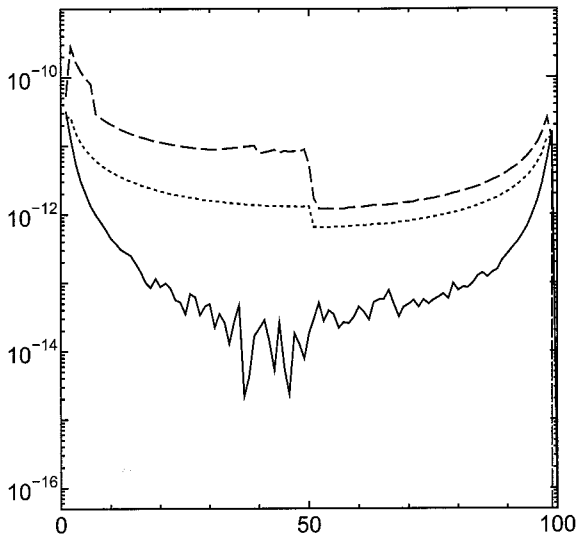

FIGURE 4.9. A graph of the suffix k for ordering singular values σ_k according to magnitude (x-axis) and relative errors in computed singular values of B_4 by the DK, the pqd and the dLV algorithms (y-axis). The dashed, dotted and solid lines are given by the DK, the pqd and the dLV algorithms, respectively.

Finally in this section we give an example of application of the Weyl type perturbation theorem (Theorem 4.4) to an estimation of error bounds of singular values computed the dLV algorithm. We restrict ourselves to the following very small example, for simplicity.

$$B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

We first compute an SVD of B by the dLV algorithm. Let us set $\delta = 1$. Then we have, for $\hat{n} = 100$, $\Sigma = \mathrm{diag}(1.80193774\mathrm{E}{-00}, 1.24697961\mathrm{E}{-00}, 4.45041868\mathrm{E}{-01})$ and

$$\hat{U} = \begin{pmatrix} 5.91009049\mathrm{E}{-01} & -7.36976230\mathrm{E}{-01} & 3.27985278\mathrm{E}{-01} \\ 7.36976230\mathrm{E}{-01} & 3.27985278\mathrm{E}{-01} & -5.91009049\mathrm{E}{-01} \\ 3.27985278\mathrm{E}{-01} & 5.91009049\mathrm{E}{-01} & 7.36976230\mathrm{E}{-01} \end{pmatrix},$$

$$\hat{V} = \begin{pmatrix} 3.27985278\mathrm{E}{-01} & -5.91009049\mathrm{E}{-01} & 7.36976230\mathrm{E}{-01} \\ 7.36976230\mathrm{E}{-01} & -3.27985278\mathrm{E}{-01} & -5.91009049\mathrm{E}{-01} \\ 5.91009049\mathrm{E}{-01} & 7.36976230\mathrm{E}{-01} & 3.27985278\mathrm{E}{-01} \end{pmatrix},$$

where the numerical test was carried out in double precision. It is to be noted that the diagonal entries σ_k of Σ are ordered according to magnitude $\sigma_1 > \sigma_2 > \sigma_3$. By using Theorem 4.4 we obtain

$$|\sigma_1 - \hat{\sigma}_1| \leq 5.0409378\text{E}{-15},$$
$$|\sigma_2 - \hat{\sigma}_2| \leq 3.9805926\text{E}{-15},$$
$$|\sigma_3 - \hat{\sigma}_3| \leq 2.4483495\text{E}{-15}$$

at the time $\hat{n} = 100$. Figure 4.10 shows a relationship between the iteration number and the estimated errors $|\sigma_k - \hat{\sigma}_k|$ of singular values. The figure shows that the dLV algorithm computes singular values at a higher precision as the iteration number n increases. The estimated error bounds are not accumulated by a large iteration number. Actual errors decrease more rapidly.

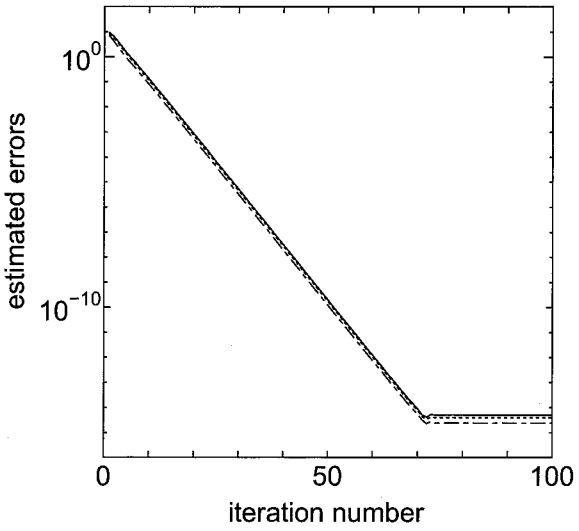

FIGURE 4.10. A graph of iteration number in the dLV algorithm (x-axis) and estimated error bounds of singular values $|\sigma_k - \hat{\sigma}_k|$ (y-axis). The solid, dotted and dashed lines correspond to the cases where $k = 1, 2$ and 3, respectively.

6. Concluding remarks

In this chapter some important features of the dLV algorithm for computing singular values of given bidiagonal matrices are revealed. By an error analysis the error bound on singular values after 1-step of the dLV algorithm is estimated. The bound is smaller than that of the DK algorithm and is as same order as that of the dqd algorithm. Forward and backward stability analyses of the dLV algorithm are also proved. Relative error bounds of singular values computed by the dLV algorithm are estimated with a help of the Weyl type perturbation theorem. A

high relative accuracy of the algorithm is then ensured. The dLV algorithm has a positivity and then the property

$$1 < 1 + \delta u_{k-1}^{(n+1)} < M$$

actually supports a high accuracy.

Numerical examples in §5 show that relative errors of singular values computed by the dLV algorithm with $\delta = 1$ are somewhat smaller than those by the DK and the pqd algorithm. The error bound and stability attributed to the dLV algorithm also hold for the sI algorithm shown in next chapter. Indeed, by introducing shifts such ill-posed bidiagonal matrices as in Cases 3 and 4 can be transformed to such well-posed matrices as in Cases 1 and 2. Consequently, the sI algorithm with variable step-size $\delta^{(n)} = 1$ is faster and more accurate than DBDSQR (without singular vector computation) in LAPACK where the shifted DK algorithm is implemented. The sI algorithm with $\delta^{(n)} = 1$ is a little bit slower but slightly more accurate than DLASQ in LAPACK where the dqds algorithm is implemented. We hope to design a new variant of the sI algorithm which is faster and more accurate than DLASQ by choosing such appropriate parameters as the discrete step-size and the shift.

CHAPTER 5

Accurate singular values and the shifted integrable schemes

1. Introduction

It is shown in Chapter 2 that the singular values of upper bidiagonal matrix B are computed by using the integrable dLV system (2.5) with arbitrary positive constant step-size $\delta > 0$. Though the convergence speed grows as δ becomes larger, numerical accuracy is deteriorated by an inappropriate choice of step-size in some cases. Moreover, in Chapter 3, a numerical algorithm for computing singular values is designed in terms of the vdLV system (3.1). In this chapter we call this algorithm the dLV algorithm, for short. The step-size $\delta^{(n)}$ of the dLV algorithm can be changed at each step n. A better choice of the flexible parameter $\delta^{(n)}$ gives a benefit from viewpoint of convergence speed and numerical accuracy. However it has not been known how to accelerate the dLV algorithm by introducing a shift of origin.

In this chapter we design a new shifted algorithm named the *shifted integrable (sI) algorithm* and compare it with LAPACK routines for computing singular values of B. From viewpoint of both convergence speed and numerical accuracy, the sI algorithm is at least four times superior to DBDSQR routine derived from the DK algorithm. The sI algorithm also runs at higher accuracy than DLASQ from the *dqd (dqds) algorithm with shift* (see §6).

The first goal in this chapter is to introduce a shift of origin into the dLV algorithm for accelerating the convergence. The second is to give a shift strategy for avoiding such a numerical instability as the *shifted qd (qds) algorithm* has. The third is to prove that the sI variable converges to some limit as $n \to \infty$. In our shifted algorithm, it is possible to find how to determine such a suitable shift that the sI varible stably converges to the shifted singular value. The property, of keeping the sI variable positve, takes an active part in the numerical stability of our shifted algorithm.

This chapter is organized as follows. In §2, we introduce new schemes and present two theorems for singular value computation of B. In §3, we show how to estimate the amount of shift so that the resulting scheme is numerically stable. In §4, we discuss the convergence of new algorithm and two particular cases where B has zero entries are described in §5. In the final section, we show test results for some examples.

2. The shifted integrable schemes

The main purpose of this section is to introduce a shift of origin into a certain recurrence relation derived from the vdLV system (3.1). Moreover we investigate an influence of the shift on singular values of upper bidiagonal matrix B.

Let us begin our analysis by introducing three mappings $\psi_j^{(n)}, j = 1,2,3$ defined by

$$\begin{aligned}
&\psi_1^{(n)} : \overline{W}^{(n)} = \{\overline{w}_k^{(n)}; k = 1,2,\ldots,2m-1\} \to U^{(n)} = \{u_k^{(n)}; k = 1,2,\cdots,2m-1\},\\
&\quad \text{such that} \quad u_k^{(n)} = \overline{w}_k^{(n)}/(1+\delta^{(n)}u_{k-1}^{(n)}), \quad u_0^{(n)} \equiv 0,\\
&\psi_2^{(n)} : U^{(n)} \to V^{(n)} = \{v_k^{(n)}; k = 1,2,\ldots,2m-1\},\\
&\quad \text{such that} \quad v_k^{(n)} = u_k^{(n)}(1+\delta^{(n)}u_{k+1}^{(n)}), \quad u_{2m}^{(n)} \equiv 0,\\
&\psi_3^{(n)} : \overline{V}^{(n)} = \{v_k^{(n)}; k = 1,2,\ldots,2m-1\} \to W^{(n+1)} = \{w_k^{(n+1)}; k = 1,2,\ldots,2m-1\},\\
&\quad \text{such that} \quad w_k^{(n+1)} = \overline{v}_k^{(n)}
\end{aligned} \tag{5.1}$$

and two bijections $\phi_j^{(n)}, j = 1,2$ defined by

$$\begin{aligned}
&\phi_1^{(n)} : W^{(n)} = \{w_k^{(n)}; k = 1,2,\ldots,2m-1\} \to \overline{W}^{(n)},\\
&\phi_2^{(n)} : V^{(n)} \to \overline{V}^{(n)},
\end{aligned} \tag{5.2}$$

for some n. The variable $u_k^{(n)}$ appeared in (5.1) corresponds to that in the vdLV system (3.1), and is equivalent to that in the dLV system (2.5) if $\delta^{(n)}$ is positive constant in n. We also regard $w_k^{(n)}$ as the variable defined by (3.4). Namely, the time evolution from n to $n+1$, by the vdLV system, is performed by using three mappings $\psi_j^{(n)}$ and two bijections $\phi_j^{(n)}$. Under the boundary condition $u_0^{(n)} \equiv 0$ and $u_{2m}^{(n)} \equiv 0$, $\psi_j^{(n)}, j = 1,2$ are also written by using a continued fraction expression as follows:

$$\begin{aligned}
\psi_1^{(n)} &: \left(\overline{w}_1^{(n)}, \frac{\overline{w}_2^{(n)}}{1+\delta^{(n)}\overline{w}_1^{(n)}}, \cdots, \left(\frac{\overline{w}_{2m-1}^{(n)}\Big|}{\Big|1} + \frac{\delta^{(n)}\overline{w}_{2m-2}^{(n)}\Big|}{\Big|1} + \cdots + \frac{\delta^{(n)}\overline{w}_2^{(n)}\Big|}{\Big|1} + \frac{\delta^{(n)}\overline{w}_1^{(n)}\Big|}{\Big|1}\right)\right)\\
&\quad \mapsto (u_1^{(n)}, u_2^{(n)}, \cdots, u_{2m-1}^{(n)}),\\
\psi_2^{(n)} &: \left(u_1^{(n)}(1+\delta^{(n)}u_2^{(n)}), \cdots, u_{2m-2}^{(n)}(1+\delta^{(n)}u_{2m-1}^{(n)}), u_{2m-1}^{(n)}\right)\\
&\quad \mapsto \left(v_1^{(n)}, \cdots, v_{2m-2}^{(n)}, v_{2m-1}^{(n)}\right).
\end{aligned}$$

Hence we see that $\psi_j^{(n)}, j = 1,2$ are bijections. It is also obvious that $\psi_3^{(n)}$ is a bijection.

Let us consider that $\psi_j^{(n)}, j = 1,2,3$ and $\phi_j^{(n)}, j = 1,2$ are defined as (5.1) and (5.2) for every n. Moreover, in this section, we assume that $w_k^{(n)} > 0, u_k^{(n)} > 0, v_k^{(n)} > 0$ and $\overline{w}_k^{(n)} > 0, \overline{v}_k^{(n)} > 0$, $k = 1,2,\ldots,2m-1$ for every n. A composite mapping

$$\psi_{vdLVs}^{(n+1)} \equiv \psi_3^{(n)} \circ \phi_2^{(n)} \circ \psi_2^{(n)} \circ \psi_1^{(n)} \circ \phi_1^{(n)} \tag{5.3}$$

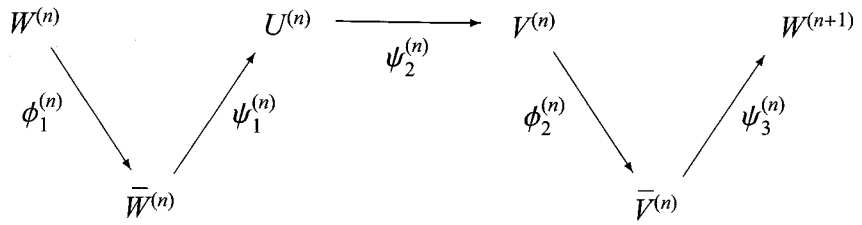

FIGURE 5.1. Evolution $W^{(n)} \to W^{(n+1)}$

produces a maping $W^{(n)} \to W^{(n+1)}$ shown as in Figure 5.1. Similarly, $\psi_1^{(n+1)} \circ \phi_1^{(n+1)} \circ \psi_3^{(n)} \circ \phi_2^{(n)} \circ \psi_2^{(n)} : U^{(n)} \to U^{(n+1)}$. Let us introduce here $\tilde{\phi}_j^{(n)}, j = 1, 2$ such that $\tilde{\phi}_1^{(n)} : w_k^{(n)} \mapsto \bar{w}_k^{(n)}$ for $k = 1, 2, \ldots, 2m-1$ and $\tilde{\phi}_2^{(n)} : v_k^{(n)} \mapsto \bar{v}_k^{(n)}$ as an example of the bijections $\phi_j^{(n)}, j = 1, 2$, respectively. Then the vdLV system can also be regarded as a dynamical system which generates an evolution from n to $n+1$ of $u_k^{(n)}$ by the composite mapping $\psi_1^{(n+1)} \circ \tilde{\phi}_1^{(n+1)} \circ \psi_3^{(n)} \circ \tilde{\phi}_2^{(n)} \circ \psi_2^{(n)} : u_k^{(n)} \mapsto u_k^{(n+1)}$.

Let us replace $\phi_1^{(n)}$ and $\phi_2^{(n)}$ in the mapping $\psi_{vdLVs}^{(n+1)}$ (5.3) with $\tilde{\phi}_1^{(n)}$ and $\tilde{\phi}_2^{(n)}$, rspectively. Then the maping $\psi_{vdLVs}^{(n+1)} : W^{(n)} \to W^{(n+1)}$ in Figure 5.1 is reduced to

$$\psi_{vdLVs}^{(n+1)} : W^{(n)} = \bar{W}^{(n)} \xrightarrow{\psi_1^{(n)}} U^{(n)} \xrightarrow{\psi_2^{(n)}} V^{(n)} = \bar{V}^{(n)} \xrightarrow{\psi_3^{(n)}} W^{(n+1)}.$$

In Chapter 2, it is shown that the singular values of

$$B^{(n)} = \begin{pmatrix} \sqrt{w_1^{(n)}} & \sqrt{w_2^{(n)}} & & \\ & \sqrt{w_3^{(n)}} & \ddots & \\ & & \ddots & \sqrt{w_{2m-2}^{(n)}} \\ 0 & & & \sqrt{w_{2m-1}^{(n)}} \end{pmatrix} \tag{5.4}$$

are invariant in n. Here the sequence $B^{(n)}$ starts from $B^{(0)} = B$ and $\psi_{vdLV}^{(n+1)} \equiv \psi_3^{(n)} \circ \tilde{\phi}_2^{(n)} \circ \psi_2^{(n)} \circ \psi_1^{(n)} \circ \tilde{\phi}_1^{(n)}$ generates an evolution $B^{(n)} \mapsto B^{(n+1)}$ where $\psi_{vdLV}^{(\infty)} \equiv \lim_{n\to\infty} \psi_{vdLV}^{(n)}$. It is also proved in Chapter 3 that $\psi_{vdLV}^{(n)} \circ \cdots \circ \psi_{vdLV}^{(1)} : (w_{2k-1}^{(0)}, w_{2k}^{(0)}) \mapsto (\sigma_k^2(B), 0)$ as $n \to \infty$, where $\sigma_k(B)$ indicates such a singular value that $\sigma_1(B) > \cdots > \sigma_m(B)$. Simultaneously, it is obvious that $\psi_{vdLV}^{(n)} \circ \cdots \circ \psi_{vdLV}^{(1)} : (w_{2k-1}^{(0)}, w_{2k}^{(0)}) \mapsto (\lambda_k(B^\top B), 0)$ as $n \to \infty$ where $\lambda_k(B^\top B)$ is the k-th eigenvalue of $B^\top B$. It is significant to note that $\lambda_k((B^{(n)})^\top B^{(n)})$ is invariant in n as long as the evolution $B^{(n)} \mapsto B^{(n+1)}$ is produced by $\psi_{vdLV}^{(n+1)}$.

It is well known in matrix eigenvalue problems [47] of that a shift of origin

$$(\overline{B}^{(n)})^\top \overline{B}^{(n)} = (B^{(n)})^\top B^{(n)} - \theta^{(n)2} I, \tag{5.5}$$

$$\overline{B}^{(n)} \equiv \begin{pmatrix} \sqrt{\overline{w}_1^{(n)}} & \sqrt{\overline{w}_2^{(n)}} & & \\ & \sqrt{\overline{w}_3^{(n)}} & \ddots & \\ & & \ddots & \sqrt{\overline{w}_{2m-2}^{(n)}} \\ 0 & & & \sqrt{\overline{w}_{2m-1}^{(n)}} \end{pmatrix}$$

is useful to accelate the convergence speed where $\theta^{(n)}$ denotes the shift at discrete time $\sum_{i=0}^{n-1} \delta^{(i)}$. We here assume that $\theta^{(n)}$ is a suitable shift for keeping $\overline{w}_k^{(n)} > 0$, $k = 1, 2, \ldots, 2m-1$. Let us introduce a parameteric bijection $\phi_{1;\theta}^{(n)}$ which is defined by

$$\phi_{1;\theta}^{(n)} : (w_{2k-2}^{(n)} + w_{2k-1}^{(n)} - \theta^{(n)2}, w_{2k-1}^{(n)} w_{2k}^{(n)}) \mapsto (\overline{w}_{2k-2}^{(n)} + \overline{w}_{2k-1}^{(n)}, \overline{w}_{2k-1}^{(n)} \overline{w}_{2k}^{(n)}) \tag{5.6}$$

with the boundary condition $w_0^{(n)} \equiv 0$ and $\overline{w}_0^{(n)} \equiv 0$. Uniquely we can compute $\overline{w}_k^{(n)}$, $k = 1, 2, \ldots, 2m-1$ from $w_k^{(n)}$ by

$$\overline{w}_{2k-1}^{(n)} = w_{2k-1}^{(n)} + w_{2k-2}^{(n)} - \theta^{(n)} + \kappa_{2k-2}^{(n)}, \quad \overline{w}_{2k-2}^{(n)} = \kappa_{2k-2}^{(n)},$$

$$\kappa_{2k-2}^{(n)} \equiv \frac{w_{2k-2}^{(n)} w_{2k-3}^{(n)}}{\left| w_{2k-3}^{(n)} + w_{2k-4}^{(n)} - \theta^{(n)2} \right|} - \frac{w_{2k-2}^{(n)} w_{2k-3}^{(n)}}{\left| w_{2k-3}^{(n)} + w_{2k-4}^{(n)} - \theta^{(n)2} \right|} - \cdots - \frac{w_2^{(n)} w_1^{(n)}}{\left| w_1^{(n)} - \theta^{(n)2} \right|}.$$

Let us replace $\phi_1^{(n)}$ and $\phi_2^{(n)}$ in (5.3) with $\tilde{\phi}_{1;\theta}^{(n)}$ and $\tilde{\phi}_2^{(n)}$, rspectively. Then $\psi_{vdLVs}^{(n+1)} : W^{(n)} \to W^{(n+1)}$ is also defined by the composite mapping $\psi_{vdLVs1}^{(n+1)} \equiv \psi_3^{(n)} \circ \tilde{\phi}_2^{(n)} \circ \psi_2^{(n)} \circ \psi_1^{(n)} \circ \phi_{1;\theta}^{(n)}$ as follows:

$$\psi_{vdLVs1}^{(n+1)} : W^{(n)} \xrightarrow{\phi_{1;\theta}^{(n)}} \overline{W}^{(n)} \xrightarrow{\psi_1^{(n)}} U^{(n)} \xrightarrow{\psi_2^{(n)}} V^{(n)} = \overline{V}^{(n)} \xrightarrow{\psi_3^{(n)}} W^{(n+1)}$$

Let $\psi_{vdLV}^{(n+1)}(X)$, $\psi_{vdLVs1}^{(n+1)}(X)$ and $\psi_{vdLVs2}^{(n+1)}(X)$, for some matrices X, denote the mappings of the enties of X by $\psi_{vdLV}^{(n+1)}$, $\psi_{vdLVs1}^{(n+1)}$ and $\psi_{vdLVs2}^{(n+1)}$, respectively, in this chapter. Since $(B^{(n+1)})^\top B^{(n+1)} = \psi_{vdLV}^{(n)}((\overline{B}^{(n)})^\top \overline{B}^{(n)}))$, we see that $\lambda_k((B^{(n+1)})^\top B^{(n+1)}) = \lambda_k((\overline{B}^{(n)})^\top \overline{B}^{(n)})$. By relating it to (5.5), it follows that

$$\lambda_k((B^{(n+1)})^\top B^{(n+1)}) = \lambda_k((B^{(n)})^\top B^{(n)}) - \theta^{(n)2}. \tag{5.7}$$

Therefore we have the following theorem for a composite mapping $\psi_{vdLVs1}^{(n+1)}$.

Theorem 5.1. $B^{(n+1)} = \psi_{vdLVs1}^{(n+1)} \circ \cdots \circ \psi_{vdLVs1}^{(1)}(B^{(0)})$ *satisfies*

$$\lambda_k((B^{(0)})^\top B^{(0)}) = \lambda_k((B^{(n+1)})^\top B^{(n+1)}) + \sum_{N=0}^{n} \theta^{(N)2}. \tag{5.8}$$

Proof. From (5.7), we have (5.8). □

In this chapter we call the procedure from $B^{(n)}$ to $B^{(n+1)}$ by the mapping $\psi^{(n+1)}_{vdLVs1}$ *the shifted integrable scheme 1.*

We here consider the case where $\phi^{(n)}_1$ is replaced by $\tilde{\phi}^{(n)}_1$ in (5.3). A composite mapping $\psi^{(n)}_3 \circ \phi^{(n)}_2 \circ \psi^{(n)}_2 \circ \psi^{(n)}_1 \circ \tilde{\phi}^{(n)}_1$ produces $W^{(n)} \to W^{(n+1)}$ that

$$W^{(n)} = \overline{W}^{(n)} \xrightarrow{\psi^{(n)}_1} U^{(n)} \xrightarrow{\psi^{(n)}_2} V^{(n)} \xrightarrow{\phi^{(n)}_2} \overline{V}^{(n)} \xrightarrow{\psi^{(n)}_3} W^{(n+1)}.$$

Simultaneouly, $\psi^{(n)}_3 \circ \phi^{(n)}_2 \circ \psi^{(n)}_2 \circ \psi^{(n)}_1 \circ \tilde{\phi}^{(n)}_1 : B^{(n)} \mapsto B^{(n+1)}$. Let us define a new mapping $\overline{\psi}^{(n)}_{3;\theta} : V^{(n)} \to W^{(n+1)}$ as

$$\overline{\psi}^{(n)}_{3;\theta} : (v^{(n)}_{2k-2} + v^{(n)}_{2k-1} - \theta^{(n)2}, v^{(n)}_{2k-1}v^{(n)}_{2k}) \mapsto (w^{(n+1)}_{2k-2} + w^{(n+1)}_{2k-1}, w^{(n+1)}_{2k-1}w^{(n+1)}_{2k}), \tag{5.9}$$

with $v^{(n)}_0 \equiv 0$, $w^{(n)}_0 \equiv 0$ and the shift $\theta^{(n)}$ which keeps $w^{(n+1)}_k > 0$, $k = 1, 2, \ldots, 2m-1$. We also see that $\overline{\psi}^{(n)}_{3;\theta}$ is a bijection since $\phi^{(n)}_1$ in (5.6) coincides with $\overline{\psi}^{(n)}_{3;\theta}$ in (5.9) by replacing $w^{(n)}_k, \overline{w}^{(n)}_k$ with $v^{(n)}_k, w^{(n+1)}_k$, respectively. Let us call the procedure from $B^{(n)}$ to $B^{(n+1)}$ by a composite mapping $\psi^{(n+1)}_{vdLVs2} \equiv \overline{\psi}^{(n)}_{3;\theta} \circ \psi^{(n)}_2 \circ \psi^{(n)}_1 \circ \tilde{\phi}^{(n)}_1$ *the shifted integrable scheme 2.* Then we have a theorem for the shifted integrable scheme 2.

Theorem 5.2. $B^{(n+1)} = \psi^{(n+1)}_{vdLVs2} \circ \cdots \circ \psi^{(1)}_{vdLVs2}(B^{(0)})$ *satisfies* (5.8).

Proof. The mapping $\psi^{(n)}_3$ is rewritten as

$$\psi^{(n)}_3 : (\overline{v}^{(n)}_{2k-2} + \overline{v}^{(n)}_{2k-1}, \overline{v}^{(n)}_{2k-1}\overline{v}^{(n)}_{2k}) \mapsto (w^{(n+1)}_{2k-2} + w^{(n+1)}_{2k-1}, w^{(n+1)}_{2k-1}w^{(n+1)}_{2k-2}). \tag{5.10}$$

Let us introduce a mapping $\phi^{(n)}_{2;\theta} : V^{(n)} \to \overline{V}^{(n)}$ given by

$$\phi^{(n)}_{2;\theta} : (v^{(n)}_{2k-2} + v^{(n)}_{2k-1} - \theta^{(n)2}, v^{(n)}_{2k-1}v^{(n)}_{2k}) \mapsto (\overline{v}^{(n)}_{2k-2} + \overline{v}^{(n)}_{2k-1}, \overline{v}^{(n)}_{2k-1}\overline{v}^{(n)}_{2k}). \tag{5.11}$$

Then we can regard $\overline{\psi}^{(n)}_{3;\theta}$ in (5.9) as a composite mapping $\psi^{(n)}_3 \circ \phi^{(n)}_{2;\theta}$. Note here that $\psi^{(n+1)}_{vdLVs2}\big|_{\theta^{(n)}=0} = \psi^{(n+1)}_{vdLV}$, since $\phi^{(n)}_{2;\theta}\big|_{\theta^{(n)}=0} = \tilde{\phi}^{(n)}_2$. Hence we see that $\lambda_k((B^{(n+1)}_{\theta^{(n)}=0})^\top B^{(n+1)}_{\theta^{(n)}=0}) = \lambda_k((B^{(n)})^\top B^{(n)})$ where $B^{(n+1)}_{\theta^{(n)}=0} \equiv B^{(n+1)}\big|_{\theta^{(n)}=0}$. Moreover a mapping $\overline{\psi}^{(n)}_{3;\theta}$ in (5.9) implies that $\lambda_k((B^{(n+1)}_{\theta^{(n)}=0})^\top B^{(n+1)}_{\theta^{(n)}=0}) = \lambda_k($ $(B^{(n+1)})^\top B^{(n+1)}) + \theta^{(n)2}$. Conseqently, $\lambda_k((B^{(n)})^\top B^{(n)}) = \lambda_k((B^{(n+1)})^\top B^{(n+1)}) + \theta^{(n)2}$. This leads to (5.8). □

3. Shift strategy

The mapping $\phi^{(n)}_1\big|_{\theta^{(n)}=0}$ in (5.6) holds $\overline{w}^{(n)}_k > 0$ if $w^{(n)}_k > 0$ for $k = 1, 2, \ldots, 2m-1$. However $\overline{w}^{(n)}_k$ is not always nonzero positive if $\theta^{(n)}$ is large. The value of $\overline{w}^{(n)}_k$ is not only negative but also numerically uncomputable in the worst case. For some k_0, if $\overline{w}^{(n)}_{2k_0-1} = 0$ by an inappropriate shift, then $\overline{w}^{(n)}_{2k_0}$ diverges to infinity, i.e. we can not compute $\overline{w}^{(n)}_{2k_0}$ numerically. Moreover we do not desire the case where $\overline{w}^{(n)}_1 > 0, \cdots, \overline{w}^{(n)}_{k_0-1} > 0, \overline{w}^{(n)}_{k_0} < 0, \cdots$ by a too large shift. This is because $1 + \delta^{(n)}u^{(n)}_k$ with $u^{(n)}_k < 0$ may be zero, i.e. $u^{(n+1)}_k$ may be numerically uncomputable by

the mapping $\psi_1^{(n)} : \bar{W}^{(n)} \to U^{(n)}$. Hence with a rather large shift the shifted integrable scheme 1 may be numerically unstable. Therefore we introduce the following proposition for keeping $\bar{w}_k^{(n)} > 0$.

Theorem 5.3. *Suppose that $w_k^{(n)} > 0$ for $k = 1, 2, \ldots, 2m-1$. Then $(B^{(n)})^\top B^{(n)}$ is positive definite symmetric. It holds that $\bar{w}_k^{(n)} > 0$ for $k = 1, 2, \ldots, 2m-1$ if and only if $\theta^{(n)2} < \lambda_m((B^{(n)})^\top B^{(n)})$, where λ_m is the minimal eigenvalue of $(B^{(n)})^\top B^{(n)}$ i.e.*

$$\theta^{(n)2} < \sigma_m^2(B^{(n)}). \tag{5.12}$$

Proof. Let $w_k^{(n)} > 0$, $k = 1, 2, \ldots, 2m-1$. Then it is obvious from (5.4) that $\sigma_k(B^{(n)}) > 0$, $k = 1, 2, \ldots, m$. Simultaneously, $\lambda_k((B^{(n)})^\top B^{(n)}) > 0$, $k = 1, 2, \ldots, m$. Hence we see that $(B^{(n)})^\top B^{(n)}$ is positive definite and symmetric.

Let us here consider the case where $\theta^{(n)2} < \lambda_m((B^{(n)})^\top B^{(n)})$. Since it is shown in §2 that $\lambda_k((\bar{B}^{(n)})^\top \bar{B}^{(n)}) = \lambda_k((B^{(n)})^\top B^{(n)}) - \theta^{(n)2}, k = 1, 2, \ldots, m$, we see that $\lambda_k((\bar{B}^{(n)})^\top \bar{B}^{(n)}) > 0, k = 1, 2, \ldots, m$, i.e. $(\bar{B}^{(n)})^\top \bar{B}^{(n)}$ is a positive definite symmetric matrix. Let $\bar{B}_k^{(n)}, k = 1, 2, \ldots, m$ denote $k \times k$ matrices defined by

$$\bar{B}_k^{(n)} = \begin{pmatrix} \sqrt{\bar{w}_1^{(n)}} & \sqrt{\bar{w}_2^{(n)}} & & \\ & \sqrt{\bar{w}_3^{(n)}} & \ddots & \\ & & \ddots & \sqrt{\bar{w}_{2k-2}^{(n)}} \\ 0 & & & \sqrt{\bar{w}_{2k-1}^{(n)}} \end{pmatrix}, \tag{5.13}$$

where $\bar{B}_m^{(n)} = \bar{B}^{(n)}$. Then the positive definite symmetric matrix $(\bar{B}^{(n)})^\top \bar{B}^{(n)}$ satisfies $\det((\bar{B}_k^{(n)})^\top \bar{B}_k^{(n)}) > 0, k = 1, 2, \ldots, m$. Note here that $\det((\bar{B}_k^{(n)})^\top \bar{B}_k^{(n)}) = \det((\bar{B}_k^{(n)})^\top)\det(\bar{B}_k^{(n)})$. Hence we derive $\prod_{j=1}^k \bar{w}_{2j-1}^{(n)} > 0, k = 1, 2, \ldots, m$, i.e. $\bar{w}_{2k-1}^{(n)} > 0, k = 1, 2, \ldots, m$. Moreover it is obvious that $\bar{w}_{2k-1}^{(n)} \bar{w}_{2k}^{(n)} = w_{2k-1}^{(n)} w_{2k}^{(n)}, k = 1, 2, \ldots, m-1$. From the assumption $w_k^{(n)} > 0, k = 1, 2, \ldots, 2m-1$, it follows that $\bar{w}_{2k}^{(n)} > 0, k = 1, 2, \ldots, m-1$.

Next we suppose $\bar{w}_k^{(n)} > 0, k = 1, 2, \ldots, 2m-1$. Then $\prod_{j=1}^k \bar{w}_{2j-1}^{(n)} > 0, k = 1, 2, \ldots, m$, i.e. $\det((\bar{B}_k^{(n)})^\top \bar{B}_k^{(n)}) > 0, k = 1, 2, \ldots, m$. Since $(\bar{B}^{(n)})^\top \bar{B}^{(n)}$ is positive definite and symmetric, we see that $\lambda_k((\bar{B}^{(n)})^\top \bar{B}^{(n)}) > 0, k = 1, 2, \ldots, m$. Note here that $\lambda_k((\bar{B}^{(n)})^\top \bar{B}^{(n)}) = \lambda_k((B^{(n)})^\top B^{(n)}) - \theta^{(n)2}, k = 1, 2, \ldots, m$. Hence it follows that $\theta^{(n)2} < \lambda_m((B^{(n)})^\top B^{(n)})$. Therefore it is concluded that $\bar{w}_k^{(n)} > 0, k = 1, 2, \ldots, 2m-1$ if and only if $\theta^{(n)2} < \lambda_m((B^{(n)})^\top B^{(n)})$, i.e. $\theta^{(n)2} < \sigma_m^2(B^{(n)})$. □

The Gershgorin-type lower bound proposed by C. R. Johnson [**19**] helps us to estimate $\sigma_m(B^{(n)})$ in (5.12) as follows:

$$\sigma_m(B^{(n)}) \geq \max\left\{0, \vartheta_1^{(n)}\right\}, \quad \vartheta_1^{(n)} \equiv \min_k \left\{ \sqrt{w_{2k-1}^{(n)}} - \frac{1}{2}\left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right) \right\}. \tag{5.14}$$

Combining it with Theorem 5.3, we have a shift strategy for avoiding numerical instability in the shifted integrable scheme 1.

Theorem 5.4. *Suppose that the initial data is as* $w_k^{(0)} > 0$ *for* $k = 1, 2, \ldots, 2m-1$ *and* ε *is some small positive constant. Then*

$$\theta^{(n)2} = \max\{0, \vartheta_1^{(n)2} - \varepsilon\} \tag{5.15}$$

is a safe choice for numerical stability in the shifted integrable scheme 1.

Moreover we consider a different shift stratagy from Theorem 5.4. Let us introduce a new variable

$$\vartheta_2^{(n)2} = \frac{1}{2} \min_k \left\{ w_{2k-1}^{(n)} - \left(w_{2k-2}^{(n)} + w_{2k}^{(n)} \right) \right\}. \tag{5.16}$$

Then we obtain the following theorem.

Theorem 5.5. *If* $\theta^{(n)2}$ *is computed by*

$$\theta^{(n)2} = \max\{0, \vartheta_2^{(n)2} - \varepsilon\}, \tag{5.17}$$

instead of (5.15)*, then the shifted integrable scheme 1 is also always numerically stable.*

Proof. Let us consider two cases $\vartheta_1^{(n)} \le 0$ and $\vartheta_1^{(n)} > 0$.

For $x, y \ge 0$, it is well known that $(x+y)/2 \ge \sqrt{xy}$. Note that $w_k^{(n)} > 0$, $k = 1, 2, \ldots, 2m-1$. Then we have

$$\begin{aligned}
\sqrt{w_{2k-1}^{(n)}}\vartheta_1^{(n)} &= \sqrt{w_{2k-1}^{(n)}} \min_k \left\{ \sqrt{w_{2k-1}^{(n)}} - \frac{1}{2}\left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right) \right\} \\
&= \min_k \left\{ w_{2k-1}^{(n)} - \sqrt{w_{2k-1}^{(n)} \cdot \frac{1}{4}\left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right)^2} \right\} \\
&\ge \min_k \left\{ \frac{1}{2} w_{2k-1}^{(n)} - \frac{1}{8}\left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right)^2 \right\} \\
&= \frac{1}{2} \min_k \left\{ w_{2k-1}^{(n)} - \frac{1}{4}\left(w_{2k-2}^{(n)} + w_{2k}^{(n)} \right) - \frac{1}{2}\sqrt{w_{2k-2}^{(n)} w_{2k}^{(n)}} \right\} \\
&\ge \frac{1}{2} \min_k \left\{ w_{2k-1}^{(n)} - \frac{1}{2}\left(w_{2k-2}^{(n)} + w_{2k}^{(n)} \right) \right\} \\
&> \vartheta_2^{(n)2}
\end{aligned}$$

which implies that $\vartheta_2^{(n)2} < 0$ if $\vartheta_1^{(n)} \le 0$. Hence $\max\{0, \vartheta_1^{(n)2} - \varepsilon\} = \max\{0, \vartheta_2^{(n)2} - \varepsilon\} = 0$ if $\vartheta_1^{(n)} \le 0$.

Suppose that $\vartheta_1^{(n)} > 0$, then it follows that

$$\begin{aligned}
\vartheta_1^{(n)2} &= \min_k \left\{ \left(\sqrt{w_{2k-1}^{(n)}} - \frac{1}{2}\left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right) \right)^2 \right\} \\
&= \min_k \left\{ w_{2k-1}^{(n)} + \frac{1}{4}\left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right)^2 - \sqrt{w_{2k-1}^{(n)} \left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right)^2} \right\} \\
&\geq \frac{1}{2} \min_k \left\{ w_{2k-1}^{(n)} - \frac{1}{2}\left(\sqrt{w_{2k-2}^{(n)}} + \sqrt{w_{2k}^{(n)}} \right)^2 \right\} \\
&\geq \frac{1}{2} \min_k \left\{ w_{2k-1}^{(n)} - \frac{1}{2}\left(w_{2k-2}^{(n)} + w_{2k}^{(n)} \right) - \sqrt{w_{2k-2}^{(n)} w_{2k}^{(n)}} \right\} \\
&\geq \frac{1}{2} \min_k \left\{ w_{2k-1}^{(n)} - \left(w_{2k-2}^{(n)} + w_{2k}^{(n)} \right) \right\} \\
&= \vartheta_2^{(n)2}.
\end{aligned}$$

Therefore it is concluded that $\sigma_m^2(B^{(n)}) > \max\{0, \vartheta_1^2 - \varepsilon\} \geq \max\{0, \vartheta_2^2 - \varepsilon\}$, i.e. $\theta^{(n)}$ in (5.17) satisfies the condition $\theta^{(n)2} < \sigma_m^2(B)$. □

One of the fortunate characteristics in (5.17) is that any square root computation does not appear at every n. In the case where we compute $\theta^{(n)2}$ by (5.15), it is necessary to compute the square root of $w_k^{(n)}$, $k = 1, 2, \ldots, 2m-1$.

The shifted integrable scheme 2 with a rather large shift also has the similar instability to the shifted integrable scheme 1. Let us recall that $\lambda_k((B^{(n+1)})^\top B^{(n+1)}) = \lambda_k((B_{\theta^{(n)}=0}^{(n+1)})^\top B_{\theta^{(n)}=0}^{(n+1)}) - \theta^{(n)2}$. According to Theorem 5.3 we see that $w_k^{(n+1)} > 0$, $k = 1, 2, \ldots, 2m-1$ if and only if $\theta^{(n)2} < \sigma_m^2(B_{\theta^{(n)}=0}^{(n+1)})$. Since it is obvious that $\sigma_m(B_{\theta^{(n)}=0}^{(n+1)}) = \sigma_m(B^{(n)})$, the shifted integrable scheme 2 is also numerically stable if $\theta^{(n)}$ is computed by (5.15) or (5.17). Moreover we may estimate a lower bound of the minimal singular value $\sigma_m(B_{\theta^{(n)}=0}^{(n+1)})$ by

$$\sigma_m(B_{\theta^{(n)}=0}^{(n+1)}) \geq \max\{0, \vartheta_3^{(n)}\}, \quad \vartheta_3^{(n)} = \min_k \left\{ \sqrt{v_{2k-1}^{(n)}} - \frac{1}{2}\left(\sqrt{v_{2k}^{(n)}} + \sqrt{v_{2k-2}^{(n)}} \right) \right\}. \tag{5.18}$$

This is because $w_k^{(n+1)} = v_k^{(n)}$ if $\theta^{(n)} = 0$ in (5.9). Similarly it follows that $\sigma_m^2(B_{\theta^{(n)}=0}^{(n+1)}) > \max\{0, \vartheta_3^{(n)2} - \varepsilon\} \geq \max\{0, \vartheta_4^{(n)2} - \varepsilon\}$ where

$$\vartheta_4^{(n)2} = \frac{1}{2}\left\{ v_{2k-1}^{(n)} - \left(v_{2k-2}^{(n)} + v_{2k}^{(n)} \right) \right\}. \tag{5.19}$$

The following theorem suggests how to determine $\theta^{(n)}$ for avoiding numerical instability in the shifted integrable scheme 2.

Theorem 5.6. *Numerical stability is always kept in the shifted integrable scheme 2 with the shift* $\theta^{(n)2} = \max\{0, \vartheta_j^{(n)2} - \varepsilon\}$ *for some* $j = 1, 2, 3, 4$.

4. Convergence to shifted singular value

In this section we consider the asymptotic behaviour of $w_k^{(n)}$ as $n \to \infty$. Moreover we explain a relationship between the limit of $w_{2k-1}^{(n)}$ as $n \to \infty$ and the singular values of $B^{(0)}$ owing to the sequence of shifts in Theorem 5.6. Let us introduce two lemma for $w_k^{(n)}$ given by both $\psi_{vdLVs1}^{(n+1)} \circ \cdots \circ \psi_{vdLVs1}^{(1)}(w_k^{(0)})$ and $\psi_{vdLVs2}^{(n+1)} \circ \cdots \circ \psi_{vdLVs2}^{(1)}(w_k^{(0)})$.

Lemma 5.7. *Let M_1 be some positive constant. Then $0 < w_k^{(n+1)} < M_1$ and $0 < u_k^{(n)} < M_1$, for all n, if $0 < w_k^{(0)} < M_1$.*

Proof. It is proved in the previous section that $0 < w_k^{(n+1)}$. In Theorems 5.1 and 5.2, we see that $\mathrm{trace}((B^{(0)})^\top B^{(0)}) = \mathrm{trace}((B^{(n+1)})^\top B^{(n+1)}) + m(\theta^{(0)^2} + \cdots + \theta^{(n)^2})$. Theorem 5.3 implies that $0 \le \theta^{(0)^2} + \cdots + \theta^{(n)^2} < \sigma_1^2(B^{(0)})$. Hence $0 < \mathrm{trace}((B^{(n+1)})^\top B^{(n+1)}) < M_2$, i.e. $0 < w_1^{(n+1)} + w_2^{(n+1)} + \cdots + w_{2m-1}^{(n+1)} < M_2$, where M_2 is some positive constant. Therefore it follows that $0 < w_k^{(n+1)} < M_1$. Since it is obvious that $u_k^{(n)} \le w_k^{(n)}$, we also have $0 < u_k^{(n)} < M_1$. □

Lemma 5.8. *Let $\gamma_{2k-1}^{(N)} \ge 1$ and $0 < \gamma_{2k}^{(N)} \le 1$ for all N. Then $w_k^{(n+1)}$ is given by*

$$w_k^{(n+1)} = \prod_{N=0}^{n} \left(\frac{1}{\gamma_k^{(N)}} \cdot \frac{1 + \delta^{(N)} u_{k+1}^{(N)}}{1 + \delta^{(N)} u_{k-1}^{(N)}} \right) w_k^{(0)}, \tag{5.20}$$

where $u_k^{(N)}$ satisfy $0 < u_k^{(N)} < M_1$.

Proof. (i) Let $w_k^{(n+1)} = \psi_{vdLVs1}^{(n+1)} \circ \cdots \circ \psi_{vdLVs1}^{(1)}(w_k^{(0)})$. Then $w_k^{(n)} = \gamma_k^{(n)} \overline{w}_k^{(n)}$ for some constants $\gamma_k^{(n)}, k = 1, 2, \ldots, 2m-1$, since it is obvious that $w_{2k-1}^{(n)} \ge \overline{w}_{2k-1}^{(n)}$ and $w_{2k}^{(n)} \le \overline{w}_{2k}^{(n)}$ in (5.6). Hence, in the mapping $\psi_{vdLVs1}^{(n+1)}$, we derive a time evolution from n to $n+1$ of $w_k^{(n)}$ as follows:

$$\begin{aligned} \frac{1 + \delta^{(n)} u_{k+1}^{(n)}}{1 + \delta^{(n)} u_{k-1}^{(n)}} w_k^{(n)} &= \gamma_k^{(n)} \frac{1 + \delta^{(n)} u_{k+1}^{(n)}}{1 + \delta^{(n)} u_{k-1}^{(n)}} \overline{w}_k^{(n)} \\ &\overset{\psi_1^{(n)}}{\longmapsto} \gamma_k^{(n)} (1 + \delta^{(n)} u_{k+1}^{(n)}) u_k^{(n)} \overset{\psi_2^{(n)}}{\longmapsto} v_k^{(n)} \overset{\psi_3^{(n)}}{\longmapsto} \gamma_k^{(n)} w_k^{(n+1)}. \end{aligned}$$

(ii) Let $w_k^{(n+1)} = \psi_{vdLVs2}^{(n+1)} \circ \cdots \circ \psi_{vdLVs2}^{(1)}(w_k^{(0)})$. Then $v_{2k-1}^{(n)} \ge \overline{v}_{2k-1}^{(n)}$ and $v_{2k}^{(n)} \le \overline{v}_{2k}^{(n)}$ in (5.11) and we see that

$$\frac{1 + \delta^{(n)} u_{k+1}^{(n)}}{1 + \delta^{(n)} u_{k-1}^{(n)}} w_k^{(n)} \overset{\psi_1^{(n)}}{\longmapsto} (1 + \delta^{(n)} u_{k+1}^{(n)}) u_k^{(n)} \overset{\psi_2^{(n)}}{\longmapsto} v_k^{(n)} = \gamma_k^{(n)} \overline{v}_k^{(n)} \overset{\psi_3^{(n)}}{\longmapsto} \gamma_k^{(n)} w_k^{(n+1)}.$$

From Case (i) and (ii) it follows that

$$w_k^{(n+1)} = \frac{1}{\gamma_k^{(n)}} \frac{1 + \delta^{(n)} u_{k+1}^{(n)}}{1 + \delta^{(n)} u_{k-1}^{(n)}} w_k^{(n)}.$$

Consequently, we have (5.20). □

It is significant to emphasize that the time evolution from n to $n+1$ of $w_k^{(n)}$ given by $\psi_{vdLVs1}^{(n+1)}$ has the same properties shown in Lemmas 5.7 and 5.8 as those of the time evolution given by $\psi_{vdLVs2}^{(n+1)}$. Moreover Lemmas 5.7 and 5.8 lead to the following proposition on the asymptotic behaviour of $w_k^{(n)}$ as $n \to \infty$.

Proposition 5.9. *As $n \to \infty$, $w_{2k-1}^{(n)} \to c_k, w_{2k}^{(n)} \to 0$, where c_k denote some positive limit and $c_1 > c_2 > \cdots > c_m$.*

Proof. Let $\bar{p}_k, p_k, s_k$ and M_3 be some positive constants. When $k = 2m-1$ in (5.20), we have $w_{2m-1}^{(n+1)} = w_{2m-1}^{(0)} / \prod_{N=0}^{n} \gamma_{2m-1}^{(N)}(1 + \delta^{(N)} u_{2m-2}^{(N)})$ which implies that $w_{2m-1}^{(0)} \geq w_{2m-1}^{(1)} \geq \cdots w_{2m-1}^{(n)} \geq \cdots$. It is proved in Lemma 5.7 that $0 < w_{2m-1}^{(n+1)} < M_1$ for all n. Since $w_{2m-1}^{(n)}$, $n = 0, 1, \ldots$, is monotonically decreasing, we see that $w_{2m-1}^{(n)} \to c_m$ as $n \to \infty$. Simultaneously, $\prod_{N=0}^{\infty} \gamma_{2m-1}^{(N)}(1 + \delta^{(N)} u_{2m-2}^{(N)}) = \bar{p}_{m-1}$. It is obvious that $1 < \prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2m-2}^{(N)}) \leq \gamma_{2m-1}^{(0)} \prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2m-2}^{(N)}) \leq \cdots \leq \prod_{N=0}^{\infty} \gamma_{2m-1}^{(N)} \prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2m-2}^{(N)})$. Hence we derive $\prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2m-2}^{(N)}) = p_{m-1}$.

Suppose that $\prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2k}^{(N)}) = p_k$. Then we see from (5.20) that $w_{2k-1}^{(0)} p_k / \prod_{N=0}^{n} \gamma_{2k-1}^{(N)}(1 + \delta^{(N)} u_{2k-2}^{(N)})$ converges to $u_{2k-1}^{(n+1)}$ as $n \to \infty$. Note here that $0 < w_{2k-1}^{(n+1)} < M_1$ Hence it follows that $0 < \prod_{N=0}^{\infty} \gamma_{2k-1}^{(N)}(1 + \delta^{(N)} u_{2k-2}^{(N)}) < M_3$. Since $\prod_{N=0}^{n} \gamma_{2k-1}^{(N)}(1 + \delta^{(N)} u_{2k-2}^{(N)})$, $n = 1, 2, \ldots$, are monotonically increasing, we derive $\prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2k-2}^{(N)}) = \bar{p}_{k-1}$. Therefore it is concluded that $w_{2k-1}^{(n)} \to w_{2k-1}^{(0)} p_k / \bar{p}_{k-1} > 0$ as $n \to \infty$, i.e.

$$\lim_{n \to \infty} w_{2k-1}^{(n)} = c_k. \tag{5.21}$$

By using the fact that $1 < \prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2k-2}^{(N)}) \leq \gamma_{2k-1}^{(0)} \prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2k-2}^{(N)}) \leq \cdots \leq \prod_{N=0}^{\infty} \gamma_{2k-1}^{(N)} \prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2k-2}^{(N)})$, we also have $\prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2k-2}^{(N)}) = p_{k-1}$.

Note here that $\sum_{N=0}^{\infty} \delta^{(N)} u_{2k}^{(N)} = s_k$ if and only if $\prod_{N=0}^{\infty}(1 + \delta^{(N)} u_{2k}^{(N)}) = p_k$ for $\delta^{(n)} u_{2k}^{(n)} > 0$, $N = 0, 1, \ldots$. Moreover $\delta^{(n)} u_{2k}^{(n)} \to 0$ as $n \to \infty$ for any positive bounded sequence $\delta^{(n)}$, if $\sum_{N=0}^{\infty} \delta^{(N)} u_{2k}^{(N)} = s_k$. Hence it follows that

$$\lim_{n \to \infty} u_{2k}^{(n)} = 0. \tag{5.22}$$

From (5.21),(5.22) and $u_k^{(n)} = w_k^{(n)} / (1 + \delta^{(n)} u_{k-1}^{(n)})$, we derive $u_{2k-1}^{(n)} \to c_k$ and $w_{2k}^{(n)} \to 0$ as $n \to \infty$ successively. Let $\alpha_k \equiv \lim_{n \to \infty} w_{2k}^{(n+1)} / w_{2k}^{(n)}$, then by using Lemma 5.8 we have

$$\alpha_k = \frac{1 + \delta^{(\infty)} c_{k+1}}{\gamma_{2k}^{(\infty)}(1 + \delta^{(\infty)} c_k)}. \tag{5.23}$$

It is obvious that $\alpha_k > 0$. Note that $w_{2k}^{(n)}$ diverges to infinity as $n \to \infty$ if $\alpha_k \geq 1$. Let us recall here that $w_{2k}^{(n)} \to 0$ as $n \to \infty$. Then we see that $\alpha_k < 1$. Hence it follows that

$$c_{k+1} < c_k \tag{5.24}$$

since $1 + \delta^{(\infty)} c_{k+1} < \gamma_{2k}^{(\infty)}(1 + \delta^{(\infty)} c_k) \leq 1 + \delta^{(\infty)} c_k$. Consequently, it is concluded that $c_1 > c_2 > \cdots > c_m$. □

A relationship between the limit of $w_k^{(n)}$, as $n \to \infty$ and $\sigma_k^2(B^{(0)})$ is derived by using Theorem 5.1, 5.2 and Proposition 5.9. Note that $\sigma_k^2(B^{(0)}) = \lambda_k((B^{(0)})^\top B^{(0)})$. Then we have the following theorem immediately.

Theorem 5.10. *As* $n \to \infty$, $w_{2k-1}^{(n)} \to \sigma_k^2(B^{(0)}) - \sum_{N=0}^{(n-1)} \theta^{(N)2}$ *and* $w_{2k}^{(n)} \to 0$ *if* $0 < w_k^{(0)} < M_1$.

5. Normalization

In the previous sections we assume that $w_k^{(0)} > 0, k = 1, 2, \ldots, 2m-1$. Note that $w_{2k}^{(n)}$ tends to 0 as n grows. For some k_0, $w_{2k_0-1}^{(n)} = 0$ if $B^{(n)}$ in (5.4) has zero-singular value. Moreover the value of $w_k^{(n)}$ is regarded as 0 in computer if $w_k^{(n)}$ is less than the machine precision. In this section we consider two cases where $w_{2k_0}^{(n)} = 0$ and $w_{2k_0-1}^{(n)} = 0$, respectively, for some k_0.

First, let us set $w_{2k_0}^{(n)} = 0$, then $B^{(n)}$ is decomposed as

$$B^{(n)} = \begin{pmatrix} B_1^{(n)} & 0 \\ 0 & B_2^{(n)} \end{pmatrix} \tag{5.25}$$

by using two upper bidiagonal matrices $B_1^{(n)} \in \boldsymbol{R}^{k_0 \times k_0}$ and $B_2^{(n)} \in \boldsymbol{R}^{(m-k_0)\times(m-k_0)}$. Both $B_1^{(n)}$ and $B_2^{(n)}$ have nonzero positive diagonal and upper subdiagonal entries. Hence singular values of $B_1^{(n)}$ and $B_2^{(n)}$ are computed as shown in the previous sections. Therefore the singular value computation of $B^{(n)}$ can be performed by computing the singular values of $B_1^{(n)}$ and $B_2^{(n)}$.

Next we explain how to compute the singular values in the case where $w_{2k_0-1}^{(n)} = 0$. Suppose that the mapping $W^{(n)} \to \bar{W}^{(n)}$ is defined by $\phi_1^{(n)}$ with $\theta^{(n)} = 0$ in (5.6). Then it is obvious that $\lambda_k((\bar{B}^{(n)})^\top \bar{B}^{(n)}) = \lambda_k((B^{(n)})^\top B^{(n)})$. This implies that we may compute the singular values of $\bar{B}^{(n)}$ instead of those of $B^{(n)}$. Since $\bar{w}_{2k_0-1}^{(n)} \bar{w}_{2k_0}^{(n)} = w_{2k_0-1}^{(n)} w_{2k_0}^{(n)}$, we see that $\bar{w}_{2k_0-1}^{(n)} \bar{w}_{2k_0}^{(n)} = 0$. Hence we may set the value of $\bar{w}_{2k_0}^{(n)}$ arbitrarily. Let $\bar{w}_{2k_0}^{(n)} = 0$. Then $\bar{B}^{(n)}$ is decomposed as the same form as in (5.25), i.e.,

$$\bar{B}^{(n)} = \begin{pmatrix} \bar{B}_1^{(n)} & 0 \\ 0 & \bar{B}_2^{(n)} \end{pmatrix}, \tag{5.26}$$

where $\bar{B}^{(n)}$ is given by the mapping $\phi_1^{(n)}|_{\theta^{(n)}=0}$ with $w_{2k_0}^{(n)} = 0$. Consequently, we can compute the singular value of $B^{(n)}$ by performing the singular value computation of $\bar{B}_1^{(n)}$ and $\bar{B}_2^{(n)}$.

6. Test results

Tests were carried out on the same computational environment as numerical experiment in Chapter 4. As numerical examples, we consider 100×100 and 1000×1000 matrices of four types in Table 4.2. In the dLV and a new *shifted integrable (sI) routines*, we adopt the same stopping criterion as in DLASQ routine in LAPACK. We also set the variable step-size $\delta^{(n)} = 1$ for $n = 0, 1, \ldots$.

First, in the singular value computation for 100×100 matrices, we compare the sI algorithm with the dLV algorithm with respect to both computational time and numerical accuracy. Table

5.1 gives the computational time of the sI and the dLV routine. We see that the sI routine is rather faster than the dLV routine. Moreover the sI routine computes the singular values at almost same time independently of matrix type. Figure 5.2 describes relative errors $|\sigma_k - \hat{\sigma}_k|/\hat{\sigma}_k$

TABLE 5.1. Computational time of the sI and the dLV routines (sec.)

	sI	dLV
Case 1 : B_1	0.02	0.27
Case 2 : B_2	0.02	0.13
Case 3 : B_3	0.02	0.88
Case 4 : B_4	0.02	174

of the singular values σ_k of the 100×100 matrix of Case 4 computed by the dLV and the sI routines, where $\hat{\sigma}_k$ are the verified correct values. Figure 5.2 suggests that the relative errors by the sI routines are much smaller than those by the dLV routines. This seems to be because the

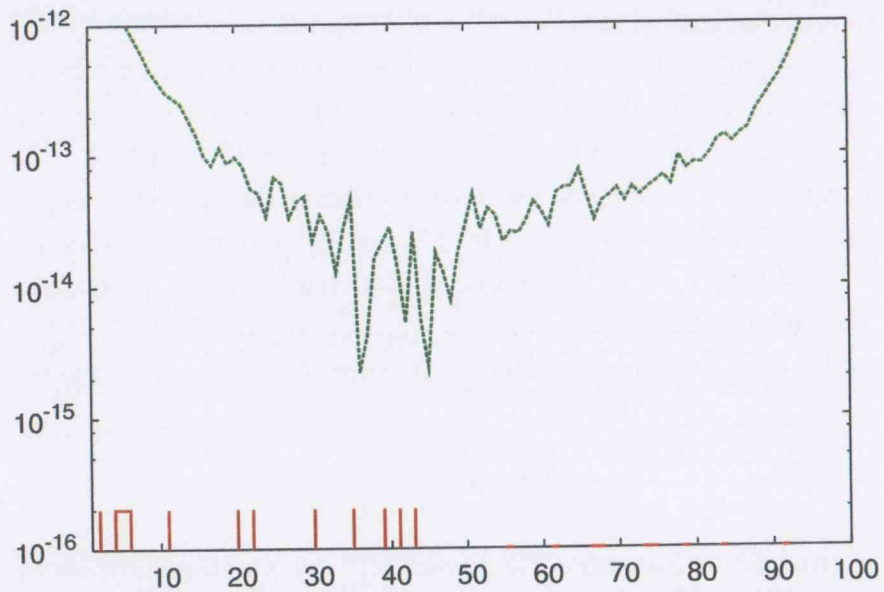

FIGURE 5.2. A graph of the suffix k for ordering singular values σ_k according to magnitude (x-axis) and relative errors in computed singular values of B_4 by the sI and dLV routines (y-axis). The red soild and green dashed lines are given by the sI and dLV routines, respectively

roundoff errors in the sI routine are less than those in the dLV routine. In other types, we also obtain the graphs similar to Figure 5.2.

Next, from viewpoint of computational time, we compare the sI routine with the DBDSQR (witout computing singular vectors) and DLASQ routines in LAPACK. Table 5.2 gives computational time of the sI, the DBDSQR and DLASQ routines in the singular value computation of

B_k, $k = 1, 2, 3, 4$, where B_k are 100×100 and 1000×1000 matrices. There is a slight difference of three routines in computational time when every B_k is 100×100. Though, in the singular value computation of 1000×1000 matrices, the sI routine computes the singular values faster than the DBDSQR routine, it does not computes them faster than the DLASQ routine.

TABLE 5.2. Computational time of the sI, the DBDSQR and the DLASQ routines (sec.)

	100×100			1000×1000		
	sI	DBDSQR	DLASQ	sI	DBDSQR	DLASQ
Case 1	0.02	0.02	0.01	1.37	2.20	0.43
Case 2	0.02	0.03	0.01	1.34	2.27	0.42
Case 3	0.02	0.03	0.01	1.32	2.43	0.42
Case 4	0.02	0.02	0.01	1.32	2.00	0.42

Finally we discuss numerical accuracy of singular values computed by the sI, the DBDSQR and the DLASQ routines for B_k, $k = 1, 2, 3, 4$ where every B_k is 100×100. Relative errors arised in the singular computation of B_4 are given by Figure 5.3. We see from Figure 5.3 that

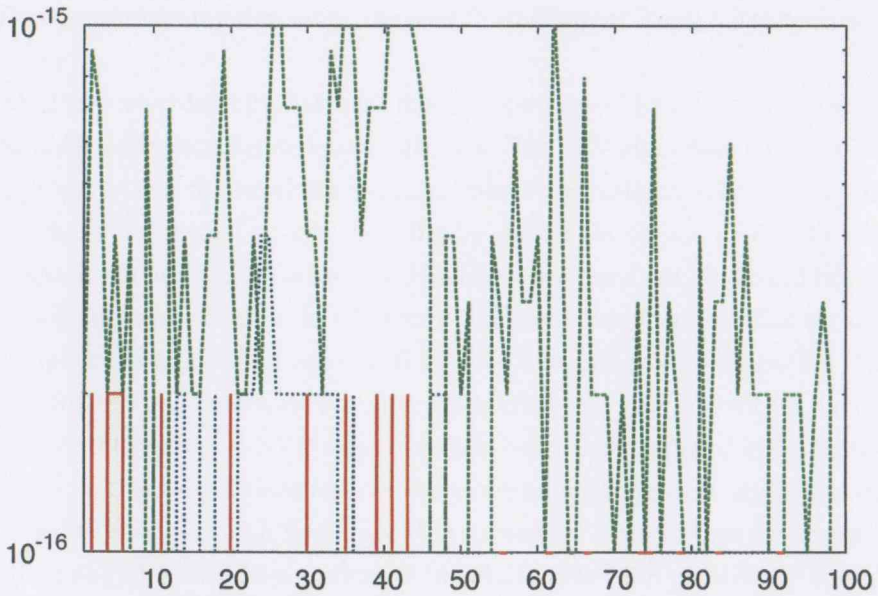

FIGURE 5.3. A graph of the suffix k for ordering singular values σ_k according to magnitude (x-axis) and relative errors in computed singular values of B_4 by the sI, DBDSQR and DLASQ routines (y-axis). The red soild, green dashed and blue dotted lines are given by the sI, DBDSQR and DLASQ routines, respectively.

the sI routine computes the singular value of B_4 at higher accuracy than the DBDSQR routine.

Hence the sI routine is superior to the DBDSQR routine with respect to both computational time and numerical accuracy. On the other hand, it is difficult to confirm a difference between the relative errors by the sI routine and those by the DLASQ routine. Let $R \equiv \sum_{k=1}^{100}((\sigma_k - \hat{\sigma}_k)/\hat{\sigma}_k)$ be the sum of relative errors. Then we have $R = 4.6 \times 10^{15}$ by the sI routine and $R = 8.5 \times 10^{15}$ by the DLASQ routine. Similar observations are given in other types. Therefore the sI routine computes the singular values at highest accuracy among three routines.

In [**39**], most of computational cost for SVD process is shown to be used for computing singular vectors. The singular value computation is a part of SVD. The SVD routine with the sI routine requires the alomost same cost as that with the DLASQ routine. Especially, in higher accurate SVD, we may use the sI routine not the LAPACK routines.

CHAPTER 6

Concluding Remarks

In this thesis, we have studied a numerical application of integrable systems to SVD algorithms. Especially, in terms of the dLV systems (with arbitrary positive discrete step-size $\delta > 0$ and varibale step-size $M > \delta^{(n)} > 0$), we have designed a new SVD algorithm.

In Chapter 2, we have shown that the dLV system with $\delta > 0$ is applicable to singular value computation. By using asymptotic expansions of Hankel determinants we have proved that the solutions expressed in Hankel determinant form of the dLV system with $\delta > 0$ converge to some limits. The vdLV system does not have a Hankel determinat solution. The convergence of the solution of the vdLV system has been shown by using a basic theory of monotonically increasing series in Chapter 3. From Lax forms of the dLV systems we have seen that those limits are the square of singular values of upper bidiagonal matrix B in Chapter 2 and 3. A new algorithm, for computing singular value, derived from Chapter 2 and 3 has been named the dLV algorithm.

In Chapter 2, we have described several basic properties of the dLV algorithm. We have seen that convergence speed is accelerated as δ increases. The dLV algorithm has been shown to have such sorting property that the resulting singular values are ordered according to magnitude. In Chapter 3, we have confirmed several benefits by a flexible choice of δ with respect to both convergence speed and numerical accuracy. However, we have not yet found how to determine the best δ at each step sequentially. In Chapter 4, we have demonstrated that the dLV algorithm computes at higher accuracy than zero-shift LAPACK routines for computing singular value. Simultaniouly, foward and backward stability analyses of the dLV algorithm have been shown. A new SVD algorithm named I-SVD algorithm has been also designed in Chapter 4.

In Chapter 5, for more acceleration, we have introduced the dLV algorithm into a shift of origin and have designed a new shifted algorithm named the sI algorithm for computing singular value. A shift stratagy for avoiding numerical instability has been presented. Though it is better than the dqds algorithm, we have not found the best staratagy of shift. We also have proved a convergence of the sI algorithm. The sI algorithm has shown to be at least in four examples superior to the LAPACK routines.

Several numerical algorithms were reconfirmed from viewpoint of integrable systems from the '90s. New algorithms have been also designed in terms of integrable systems, however, to

the best of our knowledge, they had not yet reached the practical use level in modern technologies. Our algorithm has enough performance for exceeding such established SVD algoritms as the routines in LAPACK and we hope that it will contribute for many fields.

In the near future, we should find a more effective stratagy of both step-size and shift. Introducing it into the SVD algorithm proposed in [**39**], we will design the best SVD algorithm. We will also consider several applications to problems appearing in mathematics, statistics, numerical analysis and engineering and so on.

Acknowledgments

The author firstly would like to thank all people who helped him with this thesis.

The author would like to express his deepest gratitude to his supervisor, Professor Y. Nakamura, for continuous encouragements and many invaluable instructions. He also acknowledge Professors M. Ohmiya and Y. Watanabe of Doshisha University for valuable advises and encouragements. He also thanks Dr. S. Tsujimoto, Dr. K. Koichi and Dr. Y. Minesaki for many fruitful discussions and helpful advises. The author is grateful to whole members of Nakamura Laboratory, Applied Mathematics Laboratory of Doshisha University, and Suuri Kyoshitsu of Osaka University for useful discussions and advises that enriched his study.

The author thanks Professor S. Oishi and Dr. T. Ogita of Waseda University for useful comments and discussions about the error analysis. Thanks are also due to all the members of I-SVD Project for assistance in numerical simulations. The author would like to thank to Professors M. Shimasaki and T. Nogi for giving him helpful advises and suggestions.

The author finally would like to be grateful to his parents and wife.

Bibliography

[1] M. T. Chu, A differential equation approach to the singular value decomposition of bidiagonal matrices, *Lin. Alg. Appl.*, **80**(1986), 71–79.

[2] A. K. Common and S. T. Hafez, Continued-fraction solutions to the Riccati equation and integrable lattice systems, *J. Phys. A: Math. Gen.*, **23**(1990), 455–466.

[3] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, Indexing by latent semantic analysis, *J. Soc. Info. Sci.*, **41**(1990), 391–407.

[4] P. Deift, J. Demmel, L.-C. Li and C. Tomei, The bidiagonal singular value decomposition and Hamiltonian mechanics, *SIAM J. Numer. Anal.*, **28**(1991), 1463–1516.

[5] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[6] J. Demmel and W. Kahan, Accurate singular values of bidiagonal matrices, *SIAM J. Sci. Sta. Comput.*, **11**(1990), 873–912.

[7] K. V. Fernando and B. N. Parlett, Accurate singular values and differential qd algorithms, *Numer. Math.*, **67**(1994), 191–229.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations 3rd edn.*, John Hopkins Univ. Press, Baltimore, 1996.

[9] G. H. Golub and C. Reinsch, Singular value decomposition and least squares solutions, *Numer. Math.*, **14** (1970), 403-420.

[10] P. Henrici, The quotient-difference algorithm, *Nat. Bur. Standards Appl. Math. Ser.*, **47**(1958), 23–46.

[11] P. Henrici, *Applied and Computational Analysis Vol. 1*, Wiley, New York, 1977.

[12] R. Hirota, Conserved quantities of a "random-time Toda equation", *J. Phys. Soc. Japan*, **66**(1997), 283–284.

[13] R. Hirota and S. Tsujimoto, Conserved quantities of a class of nonlinear difference equations, *J. Phys. Soc. Japan*, **64**(1995), 3125–3127.

[14] R. Hirota, S. Tsujimoto and T. Imai, Difference scheme of soliton equations, in *Future Directions of Nonlinear Dynamics in Physical and Biological Systems*, P. L. Christiansen, J. C. Eilbeck, and R. D. Parmentier, eds., Plenum, New York, 1993, pp. 7–15.

[15] M. Iwasaki and Y. Nakamura, On a convergence of solution of the discrete Lotka-Volterra system, *Inverse Problems*, **18**(2002), 1569–1578.

[16] M. Iwasaki and Y. Nakamura, An application of the discrete Lotka-Volterra system with variable step-size to singular value computation, *Inverse Problems*, **20**(2004), 553–563.

[17] M. Iwasaki, Y. Nakamura and N. Yamamoto, On discrete Lotka-Volterra algorithm for singular values: error analysis and stability, sumbitted, 2004.

[18] M. Iwasaki and Y. Nakamura, Accurate computation of singular values in terms of shifted integrable algorithm, preprint, 2004.

[19] C. R. Johnson, A Gersgorin-type lower bound for the smallest singular value, *Lin. Alg. Appl.*, **112** (1989), 1-7.

[20] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, SIAM, Philadelphia, 1974

[21] LAPACK, http://www.netlib.org/lapack/.

[22] Y. Minesaki and Y. Nakamura, The discrete relativistic Toda molecule equation and a Padé approximation algorithm, *Numer. Algorithms*, **27**(2001), 219–235.

[23] J. K. Moser, Finitely many mass points on the line under the influence of an exponential potential – An integrable system –, in *Dynamical Systems. Theory and Applications*, J. Moser ed., Lec. Notes in Phys., Vol. 38, Springer-Verlag, Berlin, 1975, pp. 467–497.

[24] A. Mukaihira and Y. Nakamura, Integrable discretization of the modified KdV equation and applications, *Inverse Problems*, **16**(2000), 413–424.

[25] A. Mukaihira and Y. Nakamura, Schur flow for orthogonal polynomials on the unit circle and its integrable discretization, *J. Comput. Appl. Math.*, **139**(2002), 75–94.

[26] A. Nagai and J. Satsuma, Discrete soliton equations and convergence acceleration algorithms, *Phys. Lett. A*, **209**(1995), 305–312.

[27] Y. Nakamura, The BCH-Goppa decoding as a moment problem and a tau-function over finite fields, *Rhys. Lett. A*, **223**(1996), 75–81.

[28] Y. Nakamura, Calculating Laplace transforms in terms of the Toda molecule, *SIAM J. Sci. Comput.*, **20**(1999), 306–317.

[29] Y. Nakamura and T. Hashimoto, On the discretization of the three-dimensional Volterra system, *Phys. Lett.*, **193**(1994), 42–46.

[30] S. Oishi, Fast enclosure of matrix eigenvalues and singular values via rounding mode controlled computation, *Lin. Alg. Appl.*, **324**(2001), 133–146.

[31] V. Papageorgiou, B. Grammaticos and A. Ramani, Integrable lattices and convergence acceleration algorithms, *Phys. Lett.*, **179**(1997), 111–115.

[32] B. N. Parlett, The new qd algorithm, *Acta Numerica*, 1995, pp. 459–491.

[33] B. N. Parlett, The QR algorithm, *Comput. Sci. Eng.*, **2**(2000), 38–42.

[34] B. N. Parlett and O. A. Marques, An implementation of the dqds algorithm (positive case), *Lin. Alg. Appl.*, **309**(2000), 217–259.

[35] H. Rutishauser, Der Quotienten-Differenzen-Algorithmus, *Z. Angre. Math. Mech.*, **5**(1954), 233–251.

[36] H. Rutishauser, Ein infinitesimales Analogon zum Quotienten-Differenzen-Algorithmus, *Arch. Math.*, **5**(1954), 132–137.

[37] H. Rutishauser, Solution of eigenvalue problems with the LR-transformation, *Nat. Bur. Standards Appl. Math. Series*, **49**(1958), 47–81.

[38] H. Rutishauser, *Lectures on Numerical Mathematices*, Birkhauser, Boston, 1990.

[39] S. Sakano, M. Iwasaki and Y. Nakamura, A high performance algorithm for computing singular vector in terms of discrete integrable systems, (in Japanese), preprint, 2004.

[40] V. Spiridonov and A. Zhedanov, Discrete-time Volterra chain and classical orthogonal polynomials, *J Phys. A: Math*, **30**(1997), 8727-8737.

[41] B. Yu Suris, Integrable discretization for lattice system: local equations of motion and their Hamiltonian properties, *Rev. Math. Phys.* **11**(1999), 727–822.

[42] W. W. Symes, The QR algorithm and scattering for the finite nonperiodic Toda lattice, *Physica* **4D**(1982), 275–280.

[43] C. Tomasi and T. Kanade, Shape and motion from image streams under orthography–a factorization mathod, *Int. Journal of Computer Vision*, **9**(2)(1992), 137–154.

[44] S. Tsujimoto, Y. Nakamura and M. Iwasaki, The discrete Lotka-Volterra system computes singular values, *Inverse Problems*, **17**(2001), 53–58.

[45] S. Tsujimoto, Studies on Discrete Nonlinear Integrable Systems, Doctor thesis, Waseda Univ., 1997.

[46] T. Tokihiro, D. Takahashi, J. Matsukidaira and J. Satsuma, From soliton equations to integrable cellular automata through a limiting procedure, *Phys. Rev. Lett.*, **76**(1996), 3247–3250.

[47] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, 1965.

List of Authors Papers Cited in the Thesis

Original Papers

[1] Satoshi Tsujimoto, Yoshimasa Nakamura and Masashi Iwasaki, The discrete Lotka-Volterra system computes singular values, Inverse Problems **17** (2001), 53–58. (Chapter 2)

[2] Masashi Iwasaki and Yoshimasa Nakamura, On a convergence of solution of the discrete Lotka-Volterra system, Inverse Problems **18** (2002), 1569–1578. (Chapter 2)

[3] Masashi Iwasaki and Yoshimasa Nakamura, An application of the discrete Lotka-Volterra system with variable step-size to singular value computation, Inverse Problems **20** (2004), 553–563. (Chapter 3)

[4] Masashi Iwasaki, Yoshimasa Nakamura and Noriyuki Yamamoto, On discrete Lotka-Volterra algorithm for singular values: error analysis and stability, submitted. (Chapter 4)

[5] Masashi Iwasaki and Yoshimasa Nakamura, Accurate computation of singular values in terms of the shifted integrable algorithm, preprint. (Chapter 5)

[6] Shinya Sakano, Masashi Iwasaki and Yoshimasa Nakamura, A high performance algorithm for computing singular vectors in terms of discrete integrable systems, (in Japanese), preprint.